AVR JTAG ICE
用户手册

Rev3.2R

www.mcuzone.com

2005-07

第一章、JTAG ICE 相关信息

AVR JTAG接口是具有4线测试存取通道(TAP)的控制器，完全符合IEEE1149.1 规格要求。IEEE1149.1已经发展成为具有能够有效地测试电路板连通性的测试标准（边界扫描）。Atmel公司的AVR 器件扩展了它一些功能，包括能够支持完整的编程和调试。

Mcuzone提供采用标准的JTAG接口，并与ATMEL官方的JTAG ICE完全兼容的AVR调试工具，使用户能够以极低的成本对具有JTAG调试接口的目标AVR芯片进行实时在线仿真和编程。

AVR OCD（On Chip Debugging 在芯片调试）协议使用户对AVR 单片机的内部资源能够进行完全的实时监控。

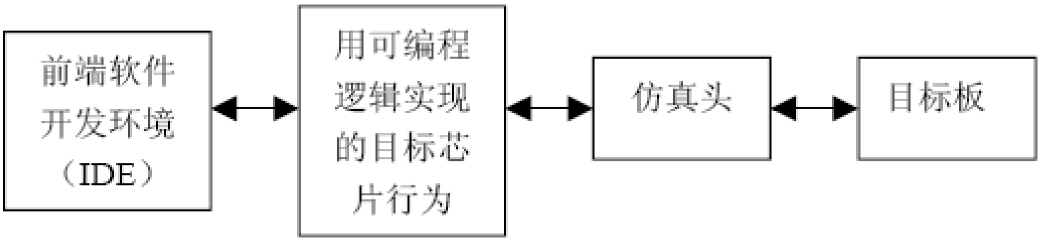
下面简单介绍一下JTAG ICE相关信息。

JTAG ICE与OCD系统：

下面简单介绍通过JTAG 接口进行在芯片调试（OCD）的概念及其与传统仿真方式之间的差异。

OCD与传统仿真方式的比较：

传统的仿真器可以对单独的或一组器件(前提是该组器件必须有很大的相似性)进行仿真。例如，我们仿真ATmega16，仿真器硬件就会设法效仿该器件的准确行为。越接近目标器件的实际动作，仿真器性能越好。下面的框图简单地描述了传统仿真方式的仿真通道。



传统的仿真器的仿真头一般价格比较昂贵，动辄几百，而且经常出现仿真不同的目标芯片需要更换不同仿真头的情况，另外在使用传统的仿真器进行仿真的时候还经常出现一个问题，就是仿真的时候一切工作正常，但是当把程序写入到目标芯片后却怎么也运行不起来，使用传统仿真器的用户一般都会遇到这个问题，此外使用传统仿真器的时候由于部分仿真器本身的原因可能会占用掉目标芯片的一部分资源，比如串口和若干的堆栈空间。

而JTAG ICE则使用一种不同的方式实现目标系统仿真。JTAG ICE能够与目标AVR片内的芯片调试系统进行接口，并通过JTAG 接口监控AVR器件的行为。

所有带有JTAG 接口的AVR器件（ATmega16及以上），都具有OCD 模块，该模块能控制目标芯片执行相关操作。所以当JTAG ICE发生一个传统的仿真行为时，JTAG ICE 将会控制目标芯片执行指令代码。下面是OCD系统的框图。

使用OCD系统，可以精确实现器件的时序和电特性。但从另一方面来说，传统的仿真器通常具有更好的可视性，可以更好地控制器件的内部资源。例如，应用AVR OCD系统就不能实现类似象跟踪缓冲器这样的功能。

另外使用由于OCD的性能限制，在使用AVR JTAG ICE进行调试的时候对于某些资源的访问有限制，具体内容将在后面讨论。

调试代码时，知道JTAG ICE 和传统的ICE 操作的不同之处是很重要的。下面将介绍一些注意事项。

1．运行模式

在该模式下，代码完全独立地在AVR器件中运行而不受JTAG ICE 的控制。JTAG ICE会不断地检查代码是否已执行到断点。当代码执行到断点时，OCD 系统会中断代码的执行，并把所有可能需要的数据（PC 指针、IO 寄存器、EEPROM、通用寄存器组和SRAM）通过JTAG 接口由JTAG ICE传送给软件开发环境（IDE）,如AVR STUDIO。由于代码独立运行不受JTAGICE 控制，所以在断点发生前JTAG ICE无法跟踪代码的执行状态。

2、停止模式

在执行到断点时，程序将被挂起，但IO的操作则仍在运行，就好象没有运行到断点时一样。假定，当断点达到时，SPI传输刚刚开始，如果使用一个传统的在线仿真器，将中断SPI的传输，如果单步执行代码，那么将这在MOSI引脚上精确地单步输出相应的位值。如果使用JTAG ICE 来做AVR 的仿真，AVR 的SPI将会全速执行完这一传输。

3、断点

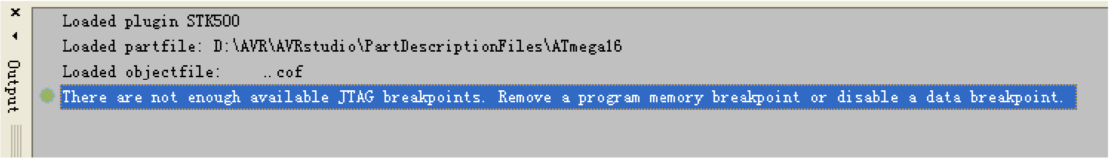
AVR 的OCD系统具有两类断点：软件断点和硬件断点

3-a、*软件断点*

软件断点是代码中的“BREAK”指令。执行这个指令时，程序将被挂起。要继续执行程序，必须从OCD逻辑中给出“start”命令。值得注意的是，并非所有的AVR 器件都支持“BREAK”指令。当在AVR Studio 中添加一条断点指令时，这条指令也将被编译并物理地写入到AVR的闪存中。当执行到这条指令时，程序将会中断。

3-b、***硬件断点***

在OCD系统中有四个寄存器,任一个都可以存储一个存储器地址。JTAG ICE需要使用一个寄存器存储单步断点,其余的3 个寄存器能以不同的组合来产生有效的中断条件。由于软件断点要重写整页的flash 数据，在断点经常改变时推荐使用硬件断点.但是,正如前面所说,硬件断点仅有三个,当在IDE中设置超过三个的硬件断点时,将出现警告.



4、**寄存器的可见性（IO 寄存器）**

从IO可见性上看，JTAG ICE 有一定限制。可见性指的是察看所有的IO寄存器中的内容。当AVR 器件运行到断点时，所有IO 寄存器中的内容被读出，并显示在前端软件开发环境中。由于读某些寄存器的操作可能改变另一些寄存器中的内容，所以这些寄存器不允许被访问。（例如读USART 数据寄存器将要清RXC位）。

相关内容可以参见文末。

5、**单步调试**

当一些控制信号被使能后，某些寄存器(WDT,EEPROM等)必须在指定的时钟周期内被读写才是有效的。因为在停止模式，IO时钟和外设仍然全速运行。单步执行这样的代码则无法得到正确的时序。因为在单步操作过程中已经执行过了无数个IO 时钟周期。为了能满足时序要求，正确地读写这些寄存器，在单步调试这样的代码时应该尽量地使用“宏”、“函数”或“执行到光标”这样的操作。

6、支持的器件

JTAG ICE 支持所有带有JTAG 接口的AVR单片机。随着新器件的不断推出，JTAG ICE也将通过升级的软件开发环境对这些新器件提供支持。一般Mcuzone在出货的时候提供当前最新版本的固件，如果日后需要更新，Mcuzone将提供永久免费升级服务。

第二章 快速入门

本章将介绍以下内容：

1，硬件环境

讲述 JTAG 和 PC 以及目标板的连接

2，软件开发环境

包含以下子内容：

2.1 AVR STUDIO 的启动

2.2 AVR JTAG ICE 属性设置

2.3 AVR JTAG ICE 的选项卡介绍

在开始使用AVR Studio前，应将目标板、AVR JTAG ICE和PC机先连接起来。为保证AVRStudio的正确自检，推荐在打开AVR Studio前应该先完成硬件连接。

一、硬件环境：

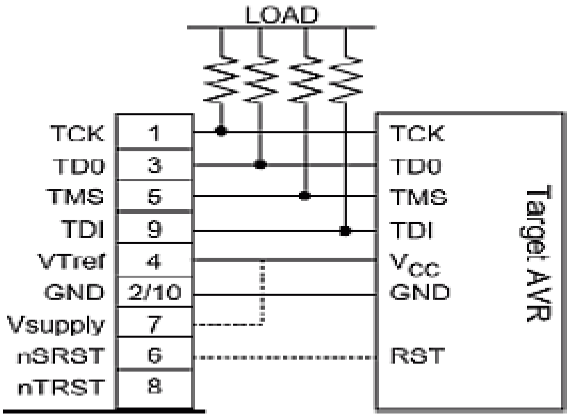
1、将JTAG ICE连接到PC机上

JTAG ICE 使用标准的RS-232口（V1和V2版本）或USB（V3**版本**）口通信。将JTAG ICE（V1**和**V2**版本**）的串口线连接到PC机的一个空闲COM端口或将JTAG ICE（V3版本）连接到PC机的USB口。AVR Studio将通过COM端口（V3采用了USB到COM的桥接设计，对AVR STUDIO而言也可以认为是一个COM口）自动检测端口上存在的JTAG ICE 仿真器。如果有其它程序占用COM 端口（如串口调试助手，超级终端等），那么在启动AVR Studio前应先关闭这些程序,否则AVR STUDIO将不能正确检测到JTAG ICE。

2、将JTAG ICE和目标电路板相连

至少有6 根线从JTAG ICE 连到目标板上，包括TCK,TDO,TDI,TMS,VTREF,GND。可选的线包括：VSUPPLY，NSRST。VSUPPLY用于通过目标板为JTAG ICE供电或者通过JTAG ICE为目标板供电（V1可以从目标板供电也可以通过外部3-5V稳压电源插头进行供电；V2板载LDO，可以从目标取电，也可以通过外接外负内正，7－9V DC的Adaptor为目标板提供3.3V电源，或者使用外部3－5V的稳压电源插头进行供电；V3直接从USB口取电，并可以为目标板提供3.3V或者5V电压）。NSRST 用于监控目标复位线。仿真时不需要该信号。但如果应用程序在MCUCSR中设置了JTD位，JTAG接口将无效。为使JTAG ICE 对目标AVR 重新编程，就需要控制复位引脚。

下面将对在目标板和JTAG ICE 间的6线连接做介绍。（在下图中用实心线画出的6 根）



上面的图指示出JTAG线如何与目标AVR 相连。为避免在线驱动出现竞争，建议在JTAG
线和外部电路间放置串连的电阻，以保证外部电路和AVR 始终工作在它们的额定电流之内。

3、JTAG ICE供电电源

不同版本的JTAG ICE具有不同的电源管理策略，V1可以通过目标板为JTAG ICE供电也可以通过板载的电源插头进行可选外部稳压电源对JTAG供电,并进一步通过跳线选择是否需要对目标板供电）；V2可以由Adaptor或者外部稳压电源插头为JTAG供电并进一步通过跳线选择是否需要对目标板供电；V3通过USB口为JTAG ICE供电，并可以进一步通过跳线为目标板提供3.3V或者5V电源。

警告:当采用两路电源分别对JTAG ICE和目标板供电的时候请先检查跳线位置！错误的设置在严重时将烧毁JTAG ICE、目标板甚至PC机端口！具体跳线设置请阅读各个版本的JTAG的用户手册。

4、目标板电源的要求

当JTAG ICE从目标板上获得电源时要注意：

1、 目标电压必须在2.7VDC-5.5VDC 范围内
2、 目标板至少要对JTAG ICE 提供150mA@3V,或者100mA@5V的电源
3、 如果不能满足这些要求，就必须将外部电源连接到JTAG ICE 上
4、 如果调试过程中，目标板电源可能会出现电压跌落，那么也必须对JTAG ICE 提供 外部电源

5、使用外部电源供电

当JTAG ICE同目标板连接好后就可以打开电源。为了保证JTAG ICE同PC机和目标板间
的通信正常，请按照以下步骤操作：

1、 打开目标板电源
2、 打开JTAG ICE电源
3、 启动AVR Studio

不同的上电顺序不会造成设备损坏,但可能引起JTAGICE仿真期间的通信异常。

注：如果目标板与仿真器采用单一电源供电则与此无关。

二、集成开发环境（IDE）

下面简单说明一下使用AVRStudio 和JTAG ICE进行在芯片仿真时的特性和注意事项。

2.1 安装AVR STUDIO:

AVR Studio 是ATMEL推出的一个面向AVR全系列单片机的官方IDE，它包含有编辑器，汇编器，调试器和ATMEL 全系列仿真器的监控程序。

最新的AVRStudio可以从http://www.atmel.com 网站下载。

注意：

由于ATMEL会对AVR STUDIO进行不定期的升级，升级AVR STUDIO的同时JTAG ICE的固件可能也会升级，故如果安装了最新的AVR STUDIO有可能需要对JTAG ICE进行固件升级，用户可按照Mcuzone指示自行升级或者直接联系Mcuzone代为升级，固件升级为免费服务。

使用JTAGICE 进行在芯片调试

在这一节中我们假定读者已经掌握了使用AVRStudio的基本知识。在此我们只针对JTAG ICE做详细介绍。关于AVRStudio的详细介绍请看AVR Studio自带的交互帮助系统。

2.1.1、连接系统

参照上文所述，将JTAGICE 与PC 及目标板连接好，依次打开目标板和JTAG ICE的电源（**上电前请一定检查跳线设置！！**）。

2.1.2、启动AVR STUDIO

在确认连接无误并打开电源后，启动AVRStudio。打开一个工程项目（Project菜单，New或者Open），在随后出现的目标窗口中选择AVR JTAG ICE和正确的目标器件，随后AVR Studio将开始寻找JTAG ICE 所在的端口。注意，当将AVR Studio的端口设置为自动时AVR Studio 是按照COM口的端口号顺序查找设备的。如果COM 口上连有其他的Ateml 设备，请先将这些设备从端口上断开。否则AVR Studio将开始寻找可支持的设备，并连接它最先找到的设备。

注意：请在启动AVR Studio之前退出所有占用COM口的程序（如：串口调试助手，超级终端等）。

JTAG ICE会检测目标板是否上电，当检测到目标板上电后，JTAG ICE就会读出目标器

件的JTAG ID，如果JTAG ICE不能读出JTAG ID，有可能是由以下原因造成的：

1、JTAG ICE没有连到目标板AVR器件上；

解决办法：检查JTAG口线连接情况。

2、目标板AVR器件的JTAG熔丝没有激活；

解决办法：使用ISP工具或编程器激活JTAG熔丝位。

3、应用程序通过MCUCSR中JTD位禁止JTAG;

解决办法：如果reset接入了JTAG ICE，可用JTAG ICE重新编程。

当JTAG ID被成功读取后，JTAG ICE会根据读到的器件ID自动识别芯片并读取目标芯

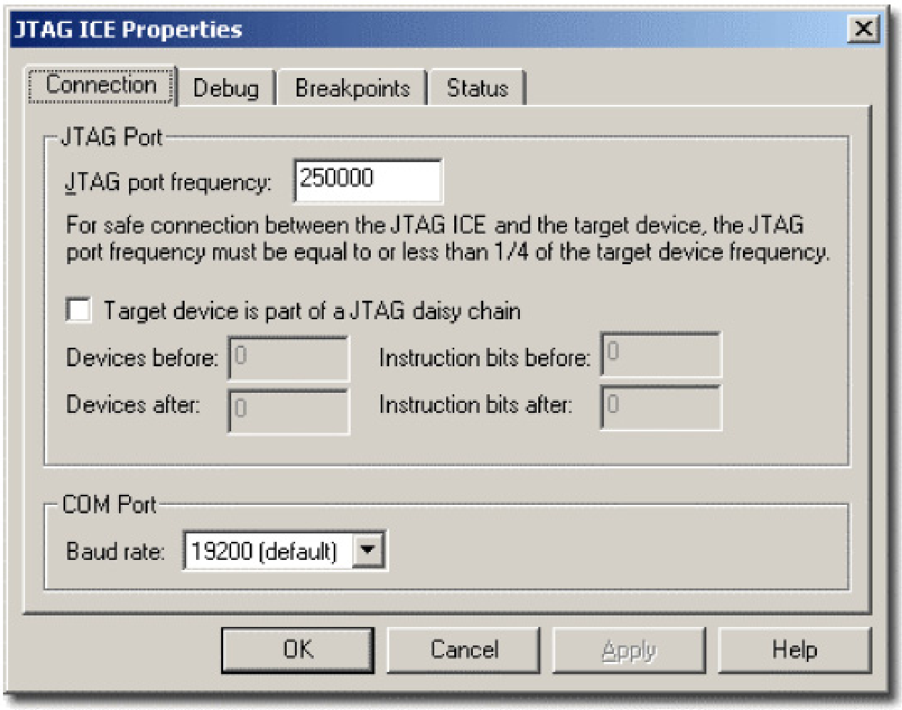
片的加密熔丝。如果加密熔丝已被编程，JTAG ICE在执行下一步操作时会擦除芯片。而且，

如果目标AVR的OCD功能已被禁止，JTAG ICE 也能自动使能目标CPU的OCD功能，退

出时可以自动禁止该项功能。

2.2、设置JTAGICE 选项

进入调试模式后，点选DEBUG 菜单->JTAG ICE Options 打开JTAGICE 选项窗口（如图）：



注意：当没有连接JTAGICE 或没有进入调试模式时，DEBUG 菜单中没有该选项

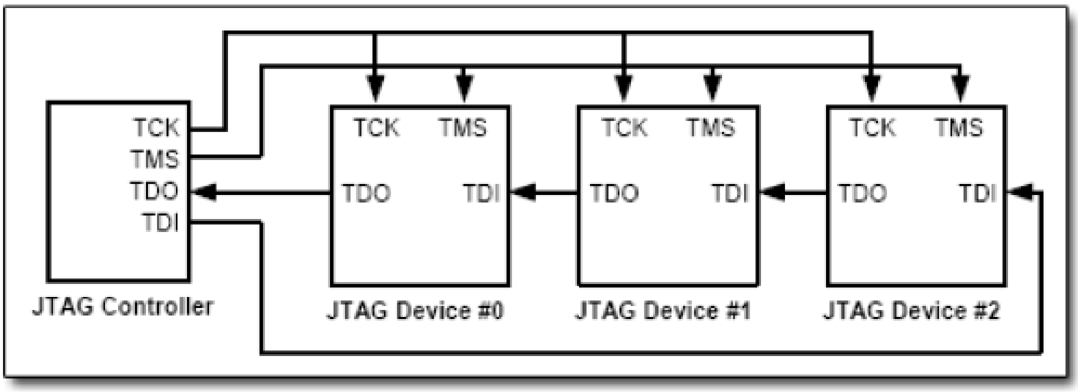
2.2.1 设置Connection选项卡：

2.2.1.1、设置JTAG 端口速率

为了保证在JTAG ICE 和目标AVR 间的正确通信，JTAG 端口的通信频率必须少于或等于1/4 的目标AVR 工作频率。如果JTAG ICE以更高的速度通信，那么虽然目标AVR 能够继续工作，但通信将会失败。另一方面，如果你指定的OCD 频率太低，也能正常通信，但会引入不必要的延时。

2.2.1.2、定位**JTAG菊花**链位置

如果目标是JTAG 菊花链的一部分，那么就必须详细说明在现行的目标前后的器件个数和指令寄存器（IR）的位长。最大的IR 长度之和是32 位。例：



上页图中共有三个器件，它们相互串联构成JTAG 链。

#0号器件位于#1号和#2号器件之前

#1号器件位于#2号器件之前，位于#0号器件之后

#2号器件位于#0号和#1号器件之后

假定各器件的指令寄存器位长如下：

#0号器件有一个16位的IR

#1号器件有一个8位的IR

#2号器件有一个16位的IR

再次假定我们要对第#0号器件做JTAG操作，那么应做如下的设置

Devices before：0

Devices after：2

Instruction bits before：0

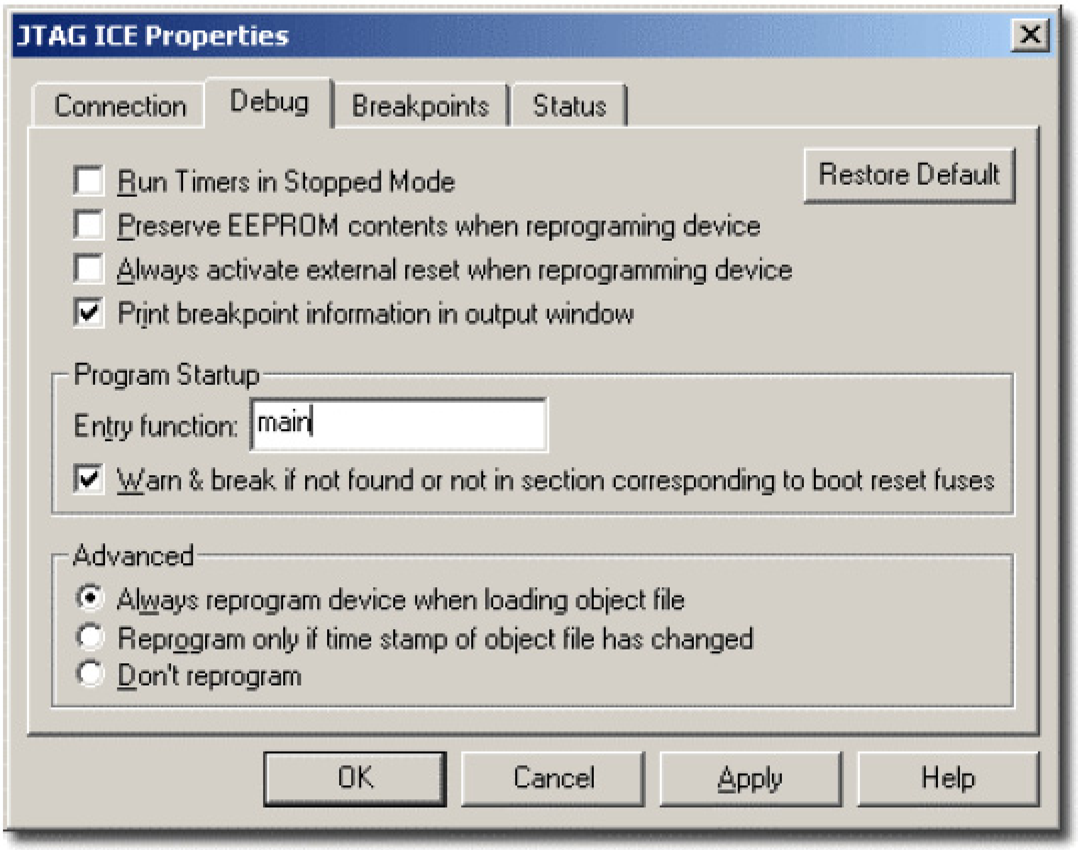
Instruction bits after：24

2.2.1.3、设置 JTAGICE 的初始化串口波特率

JTAG ICE允许用户选择初始化时的通信速率，缺省波特率设置在19200上。用户可以在9600、19200和115200中选择，但如果发生通信问题，请降低通信的波特率。

2.2.2 设置JTAGICE调试选项

点击JTAG ICE 属性窗口中的DEBUG选项卡，即可切换到调试选项设置窗口，如下图示：



2.2.2.1、“Run Times in Stopped Mode”选框

这种模式允许定时器在单步执行时，以标准速度继续运行。当这个选框没有被选中时，定时器将通过代码，准确的按单步的次数来增加计数值。

2.2.2.2、“Preserve EEPROM contents when reprogramming device”选框

该选框的值会改变目标芯片中“保护EEPROM熔丝位”的值，如果该选框被选中,那么在重编程芯片或做芯片擦除时,目标芯片内EEPROM的数据就不会被清除。

2.2.2.3、“Always activate external reset when reprogramming device”选框

该选框若被选中，那么当JTAGICE完成芯片擦除后，就会释放复位线，使外部复位有效。

2.2.2.4、“Print breakpoint information in output windows”选框

选中该选框，将在“输出窗口”打印出断点的相关信息。

2.2.2.5、设置程序入口

当执行高级复位时，AVRStudio会设法从入口处开始执行。默认的入口函数是“main”，用户也可以根据程序更改其名字。

2.2.2.6、“Warn & break if not found or not in section corresponding to boot reset fuses”选框

选中该选框，AVRStudio会检测程序入口与 Boot reset 熔丝的状态是否匹配。如果该选框被选中后，仿真一开始就出现相关警告，或PC指针指向复位向量的地址，有可能是由下列原因引起的：

入口函数没有指定，或入口函数不存在

入口函数位于应用程序区，而Boot reset 熔丝被编程

入口函数位于Boot区，而Boot reset 熔丝没有被编程

该项若被关闭，将不会产生相关警告，而且AVRStudio将尝试到达程序的入口处

2.2.2.7、高级选项

该选项是单选框,用于设置仿真目标芯片时是否对目标芯片进行重编程。

“Always reprogram device when loading object file”

选择该项,JTAGICE将会在每次装载目标文件的时候对目标芯片编程

“Reprogram only if time stamp of object files has changed”

选择该项,JTAGICE只在目标文件被更新后才对目标芯片编程

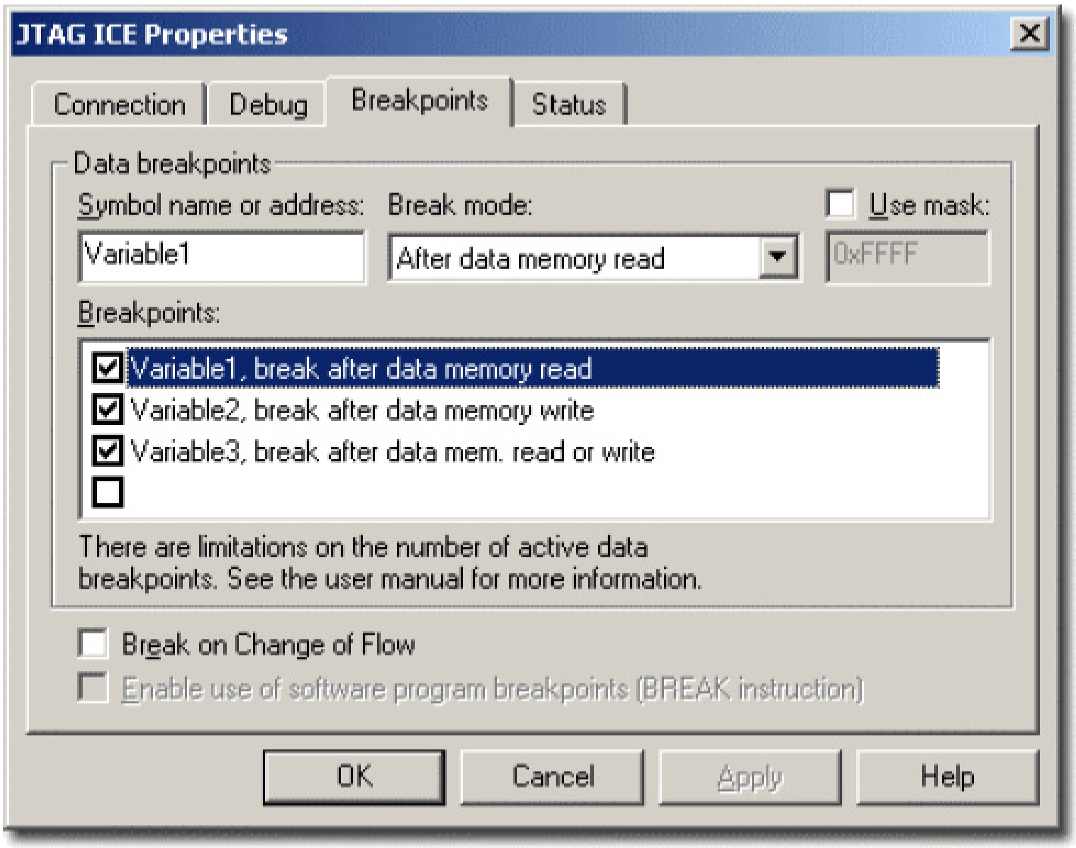
“Don’t reprogram”

选择该项,JTAGICE始终不对目标芯片编程

注:为了优化编程的速度,编程时没有校验

2.2.3 为JTAG调试设置断点

点击JTAG ICE 属性窗口中的Breakpoints选项卡，即可切换到断点设置窗口，如下图示：



2.2.3.1、硬件断点

与其他的ATMEL AVR仿真器不同的是，JTAGICE只有有限的3个通用硬件断点。这3个通用
硬件断点可做如下配置：

- 3个通用断点
- 2个通用断点和1个数据断点
- 1个通用断点和2个数据断点
- 1个通用断点和1个 SRAM 断点
- 1个通用断点和1个 FLASH 断点

“Break on change of program flow” 选框可以使能或关闭在程序执行中流程改变(PC值非线性变化)发生的中断，不管其他断点选项是否使用。

AVRStudio会不断跟踪每一种被使用的断点模式，并且在设置了多于3个硬件断点时给出相关的警告。

2.2.3.2、软件断点

某些AVR 器件支持中断指令。使用中断指令意味着在调试期间，中断指令将占用闪存中的一条真实指令的位置。通过放置中断指令，将允许使用无数的断点。因为中断指令要占用Flash 的空间，所以在增加或减少断点时，Flash 都必须重新编程。

注：中断指令比硬件断点响应的速度慢。

2.2.3.3、关于断点的三种不同的可用选项：

1. 自动使用各种断点，但硬件断点优先

在该模式下，AVRStudio 会自动分析断点，并尝试地放置硬件断点，并以最合适的方式放置中断指令。

2. 只使用硬件断点

使用该选项，那么只能设置3 个通用断点。该模式是缺省的断点模式

3. 只使用软件断点

该模式迫使JTAGICE 只使用软件断点，在该模式下硬件断点无效。

注：使用ATmega128 时，如果设置了ATmega103 兼容模式，将无法使用软件断点。这是因为ATmega103 不支持SPM 指令造成的。

2.2.3.4、通用断点

通用断点可以放置在代码的任意位置上。JTAGICE 支持原代码调试。断点可以放置在汇编或高级语言原代码中，AVR 器件将会在执行的过程中停止在被放置断点的语句前。

2.2.3.5、数据存储器断点

数据存储器断点可以设置在以下三种方式中

- 数据存储器读
- 数据存储器写
- 数据存储器读写

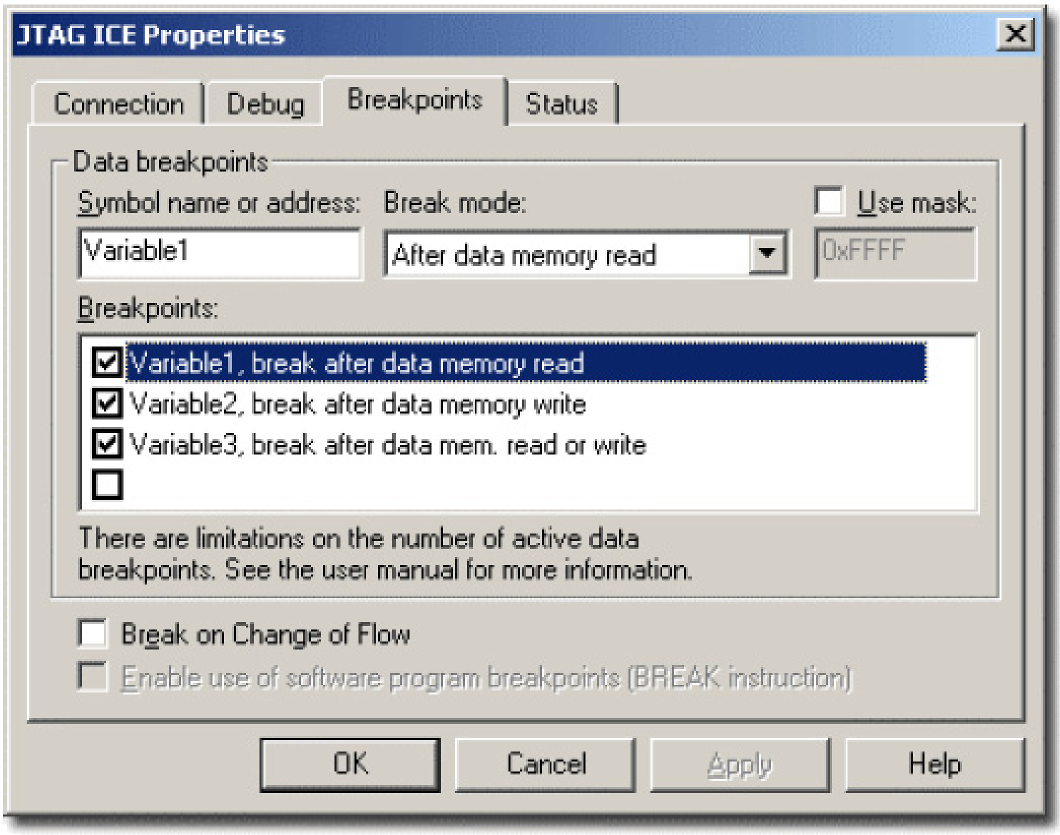
请注意，同时只能使用两个数据断点。数据断点可以设置在IO 空间或SRAM 空间但不能设置在寄存器(文件)中。为了使用变量名（比如：Variable1）做断点，目标文件中应该包含这些变量名信息。如果包含了变量名信息，那么就可以在C 编译器或汇编器中使用变量名做为数据断点（比如WinAVR、EWBAVR、ICCAVR、IAR Assembler）

注1：AVRStudio 的汇编程序不会产生变量名信息

注2：一些编译器不会为位于IO 空间的变量产生变量名信息，请使用绝对地址来定义数据中断。

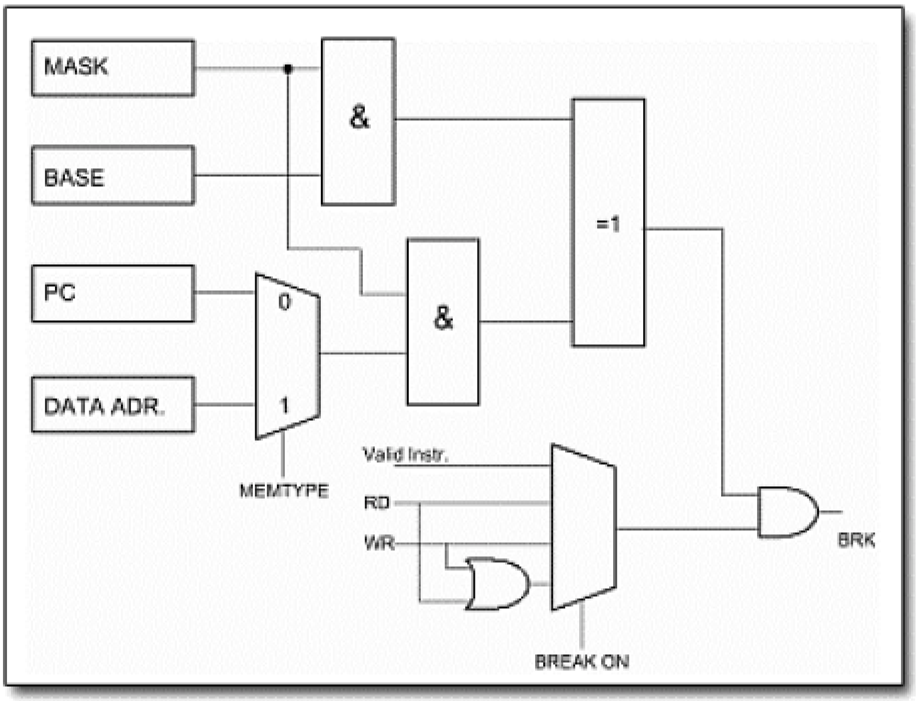
注3：使用数据中断，AVR 将会在执行了满足中断条件的语句之后中断。

数据断点设置完成后如下图示：



2.2.3.6、数据存储器的断点掩码

在上图中右上方有个选框(Use mask)，称之为断点掩码，当选中该选框，位于选框下方的输入框变为有效状态。这时必须详细指明基地址和断点掩码。这两个寄存器按位与后产生的结果与PC 值或是数据存储器的地址比较来判断是否出现了数据中断条件，实现的框图如下：



序号	基地址	掩码	位与结果	掩码0个数
1	1010 1010 0101 0101	1111 1111 1111 1111	1010 1010 0101 0101	0
2		1111 1101 1111 1111	1010 10x0 0101 0101	1
3		1111 0000 1111 1111	1010 xxxx 0101 0101	4
4		1010 1010 0101 0101	1x1x 1x1x x0x0 x0x0	8
5		0000 0000 0000 0000	xxxx xxxx xxxx xxxx	16

正如上表所示，如果掩码的某位为0，无论PC 值或数据地址寄存器相对应位的内容如何，此位任何时候都满足中断条件。要是掩码的某位为1，相应的PC值或数据地址寄存器数据位的值，必须同基地址相应位的值相等，才满足中断条件。

①上表中的例1 掩码全为高，只有基地址会产生有效中断向量。而且该中断向量是唯一的：1010 1010 0101 0101

②上表中的例2 掩码中有1 个0，将产生两个有效中断向量：

1010 1000 0101 0101

1010 1010 0101 0101

③上表中的例5 掩码全为低，意味着所有地址都是有效中断向量，AVR 将单步执行。

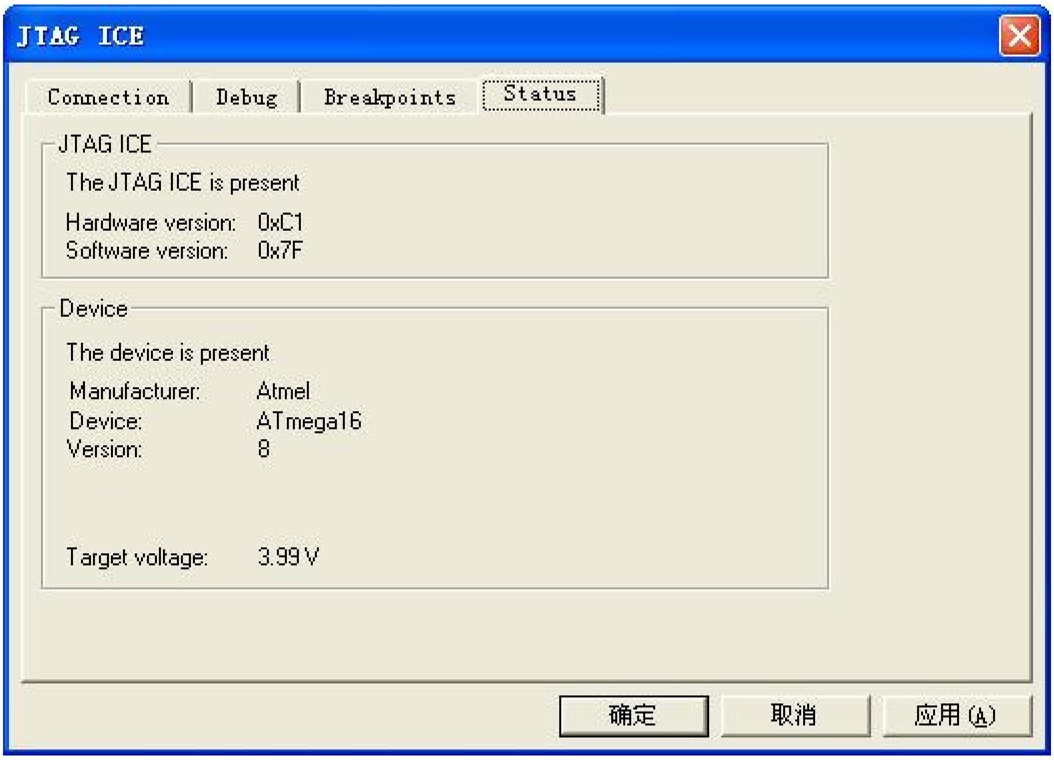
如果PC 或数据地址与一个有效中断向量匹配，控制逻辑将引起中断。AVR 将会在执行了满足中断条件的那条语句后暂停程序。

2.2.3.7、程序进入分支或**PC** 发生跳转时引起中断

JTAGICE 还可以不依赖于3 个硬件断点和中断指令来中断程序。当“Break on Change of Flow”选框有效时，PC 值不是线性变化都将引起程序中断。在这种中断下，AVR 会在执行了满足中断条件的那条语句后暂停程序。

2. 2. 4 检查JTAGICE 状态

点击JTAG ICE 属性窗口中的Status 选项卡，即可切换到状态窗口，如下图示：

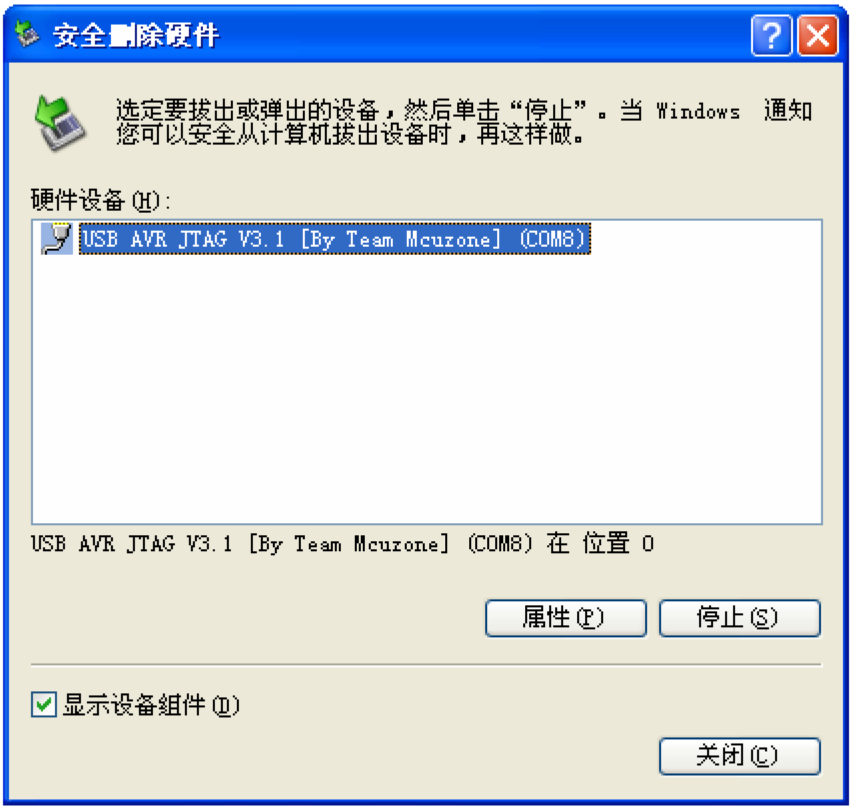


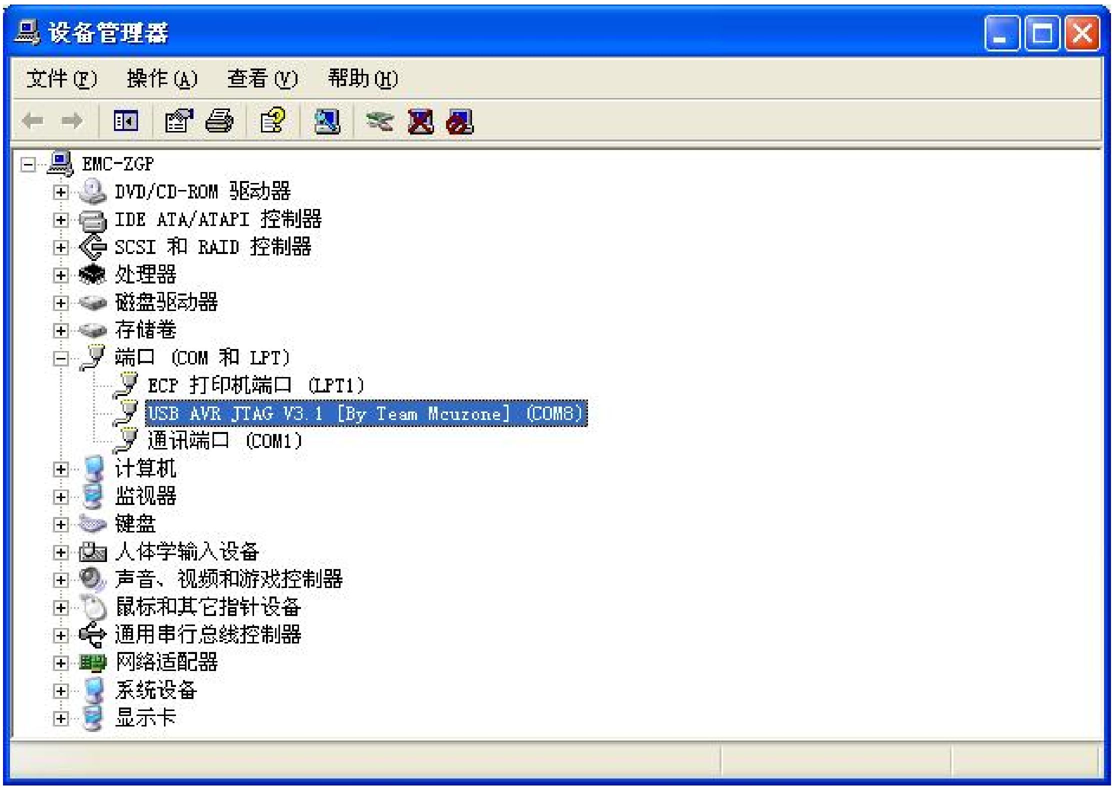
该窗口显示了JTAGICE 及目标器件的状态，在多目标芯片时，该状态窗口给出的目标芯片型号可以辅助判断JTAG 链的定位。

2.3 JTAGICE选项卡设置

JTAG ICE除了能作为在线仿真器使用外，它还可以当作编程器来使用。当作为编程器使用时,JTAGICE 可以对支持的器件的FLASH、EEPROM、熔丝位和加密位编程，可以执行芯片擦除、编程、校验、加密的操作。尤其对熔丝位的编程，使用JTAGICE 比使用通用编程器更加方便。JTAGICE 提供了熔丝组合配置的列表，用户可以直接明了地配置熔丝。在AVRStusio中点击“TOOLS->STK500/AVRISP/JTAGICE->STK500/AVRISP/JTAGICE”（4.10以前版本）或者“TOOLS->PROGRAM AVR->AUTO CONNECT”即可打开JTAGICE 编程窗口。与在线仿真时的连接要求一样，必须先退出所有占用JTAGICE 端口的程序。如果正在调试，那么也必须点击DEBUG->Stop Debugging 来退出仿真调试程序对JTAGICE的占用。

最后检查电路是否都已连通了，确认无误后即可上电，在编程模式下没有上电顺序的要求。如果这些准备工作都已经完成了，就可以打开JTAGICE 编程窗口了。如果使用USB版本（即V3版本）的JTAGICE，正确安装驱动后，系统将会虚拟出一个串口,用户可以通过控制面板中系统下的设备管理器或者双击任务栏右下角的即插即用设备得到USB JTAG ICE的设备端口号，如下图示：





USB接口的JTAG ICE采用了USB到RS232的桥接芯片。接入USB版JTAG ICE，安装完提供的驱动之后，打开设备管理器窗口，在端口中将出现“USB AVR JTAG V3.1[By Team Mcuzone]”的端口设备，紧随其后的就是该设备对应的端口号。

上图中USB JTAG ICE的端口号是COM8。

JTAGICE 编程窗口如下图所示：

在AVR Stusio中点击“TOOLS->STK500/AVRISP/JTAGICE->STK500/AVRISP/JTAGICE”（4.10以前版本）或者“TOOLS->PROGRAM AVR->AUTO CONNECT”打开JTAGICE 编程窗口。打开该窗口后如果下方信息栏中出现如下提示：

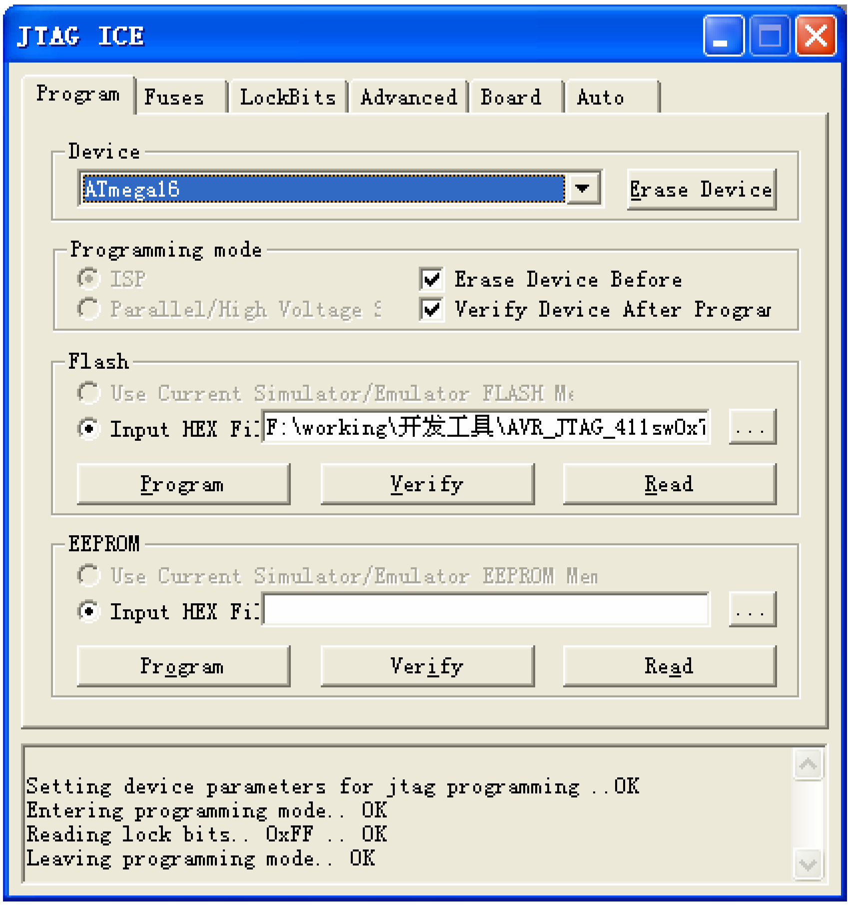
Detecting: FAILED!

那么请检查你的电源和仿真器的连接。如果你使用的是RS232 版的JTAGICE 那么请与我们联系，如果你使用的是USB 版的JTAGICE 那么先按上述的方法在控制面板里找出仿真器的设备端口号，然后打开JTAG ICE 编程窗口，点选Advanced 选项卡，将会出现上图画面。检查COM 口的设置。因为有时自动配置无法找到设备，所以建议将COM口设为JTAG ICE的设备端口号，并将波特率设为19200。然后，关闭该窗口，并重新打开它。如果依然无法检测到JTAGICE，那么也请与我们联系。

接下来，介绍一下各选项卡的功能。

1、 Program 选项卡

Program 选项卡如下图示：

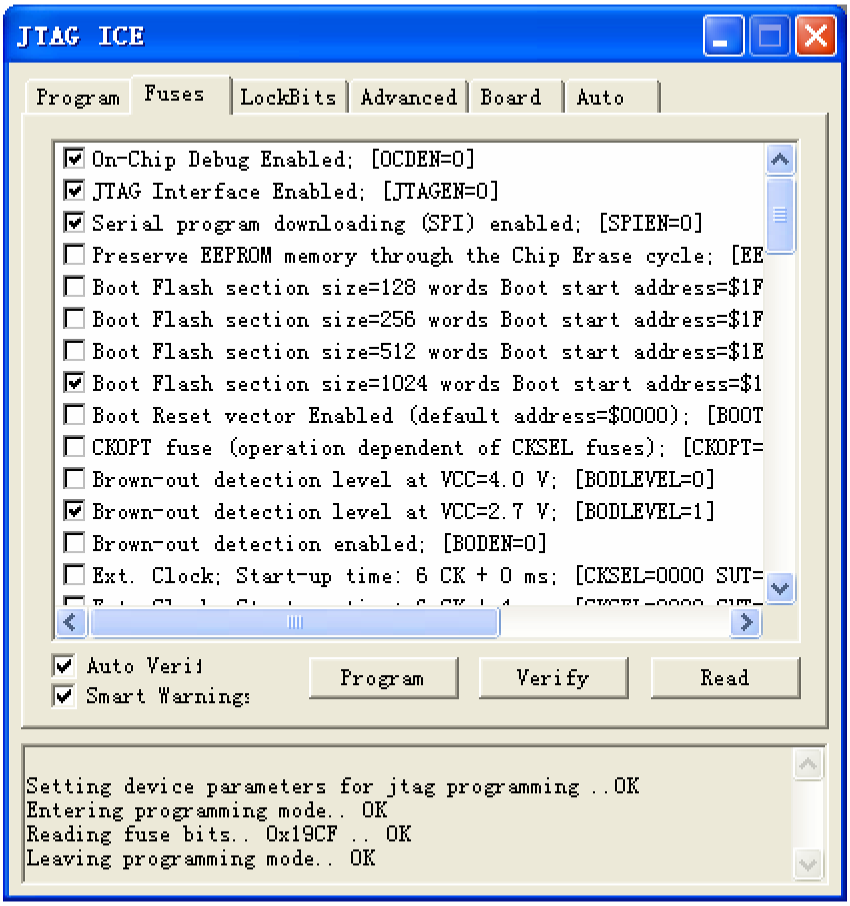


首先选择器件。注意，只有带JTAG 口功能的AVR 才支持JTAGICE 编程。为了编程的可靠，建议选中“Erase Device Before”和“Verify Device After Program”选框。保证每次对芯片烧写程序的时候，都先擦除芯片再编程并在编

程后对目标芯片做校验。

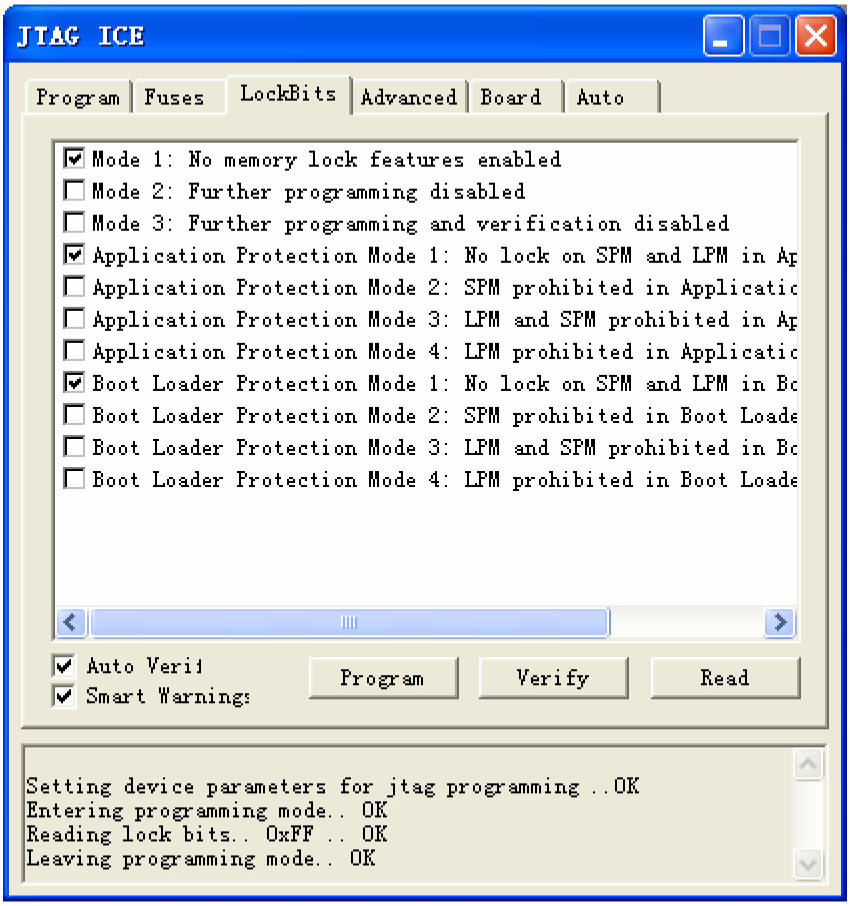
2、 Fuse 选项卡

Fuse 选项卡如下图示：



如图所示，该列表列出了所有的熔丝配置的组合。用户将会发现有些选项是独立的，有些选项是组合的。例，上图是Atmega16 的熔丝配置图。第1 到第4 项就是独立选项，而第5 到第8 就是组合选项。独立选项用于配置独立的熔丝，而组合选项用于配置需要组合的熔丝，简单地说，就是组合选项中只有一种是有效的，你无法同时选中同一熔丝组中的两种不同配置，如果你先选了第5 项，再选第6 项时，第5 项左边的选中标志会自动消失。在熔丝编程时要特别注意的是，请保证“JTAG Interface Enable”一项始终有效，如果用户禁止了该熔丝，那么目标芯片就将失去与JTAGICE 的通信能力，只有通过串行或并行编程的办法才能重新激活该熔丝位（即用通用编程器重新激活该熔丝位）。

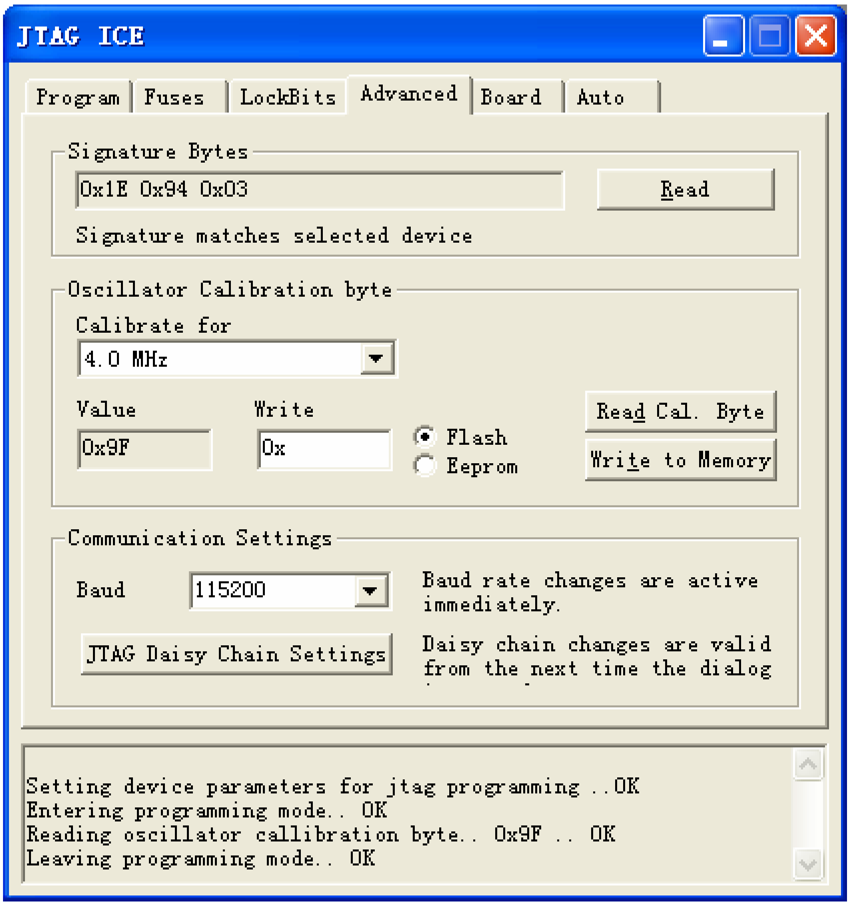
3、 LockBits 选项卡



加密位选项卡也同样列出了所有加密位的组合。当你选定了采用某级加密，并在该加密级别下对加密位进行编程，如果想要将加密位删除，那么只能对芯片做擦除操作。但是你可以在某一加密级别下对芯片做更高级别的加密操作而无须重写芯片。

4、 Advanced 选项卡

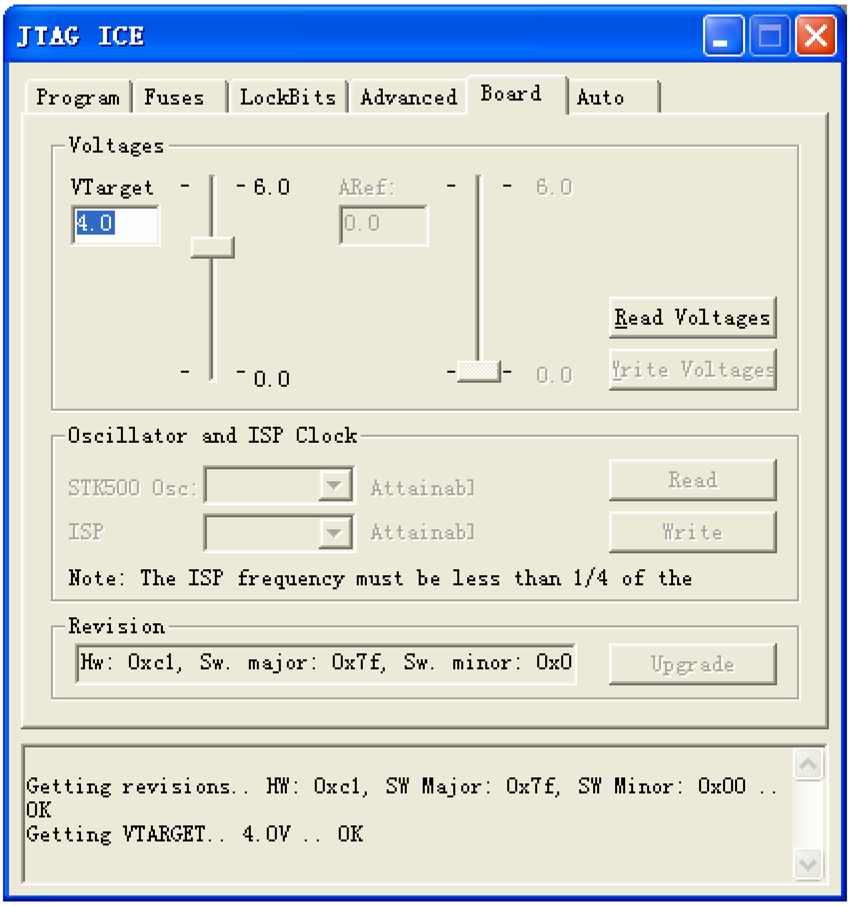
如图所示Advanced 的功能如下：



Read Signature 用于读取芯片的ID 号，该ID 号与芯片的型号是一一对应的。下面这一栏“Oscillator Calibration byte”用于读取片内振荡器的频率校准值。选择芯片的振荡频率后“Value”和“Write”都从灰色变成可用的状态，这时可以读出片内的配置值，在“write”框中用“0xXX”十六进制格式指定写入FLASH 或EEPROM 中的地址，点击“Write to Memory”就能把校准的值写入到指定的地址空间中以便程序的调用。

5、 Board 选项卡

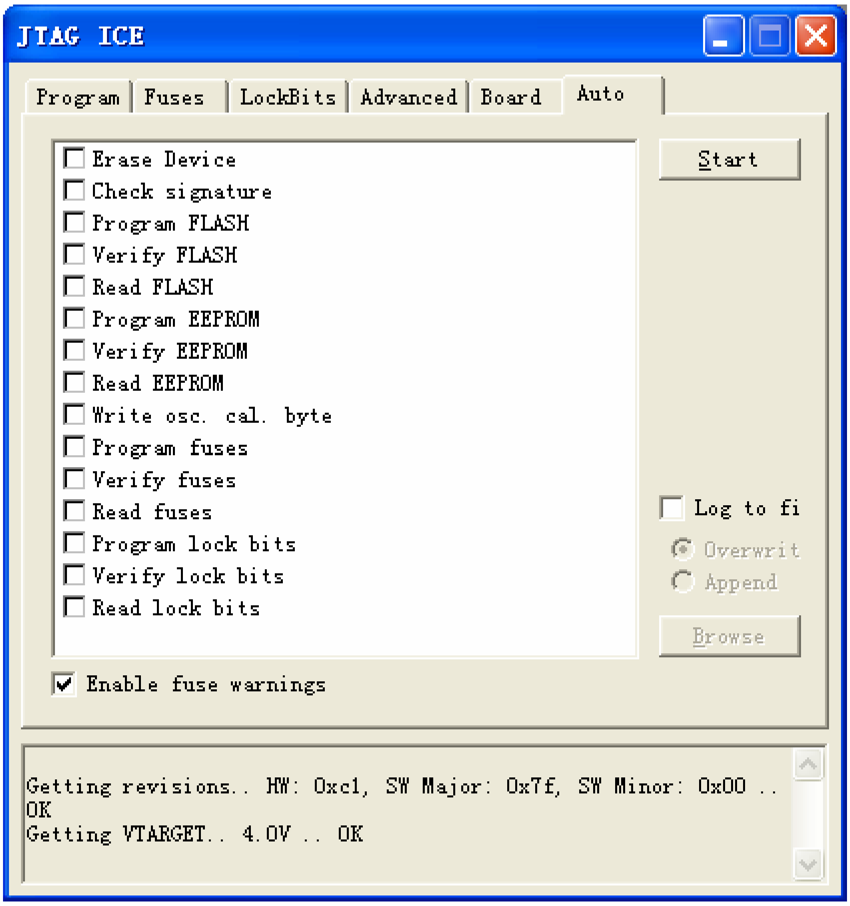
Board 选项卡没有特殊的功能，只是为用户提供一些关于目标板和JTAGICE 的信息，如下图所示：



“读取电压值”,即可读取目标板的电压值。该栏的信息还指示出JTAGICE 硬件和软件的版本号。

6、 Auto 选项卡

Auto 选项卡为用户在JTAGICE 上实现了类似于编程器的自动编程功能，如下图所示：


JTAG ICE
Program
Fuses
LockBits
Advanced
Board
Auto
Erase Device
Check signature
Program FLASH
Verify FLASH
Read FLASH
Program EEPROM
Verify EEPROM
Read EEPROM
Write osc. cal. byte
Program fuses
Verify fuses
Read fuses
Program lock bits
Verify lock bits
Read lock bits
Start
Log to fi
Overwrit
Append
Browse
Enable fuse warnings
Getting revisions.. HW: 0xc1, SW Major: 0x7f, SW Minor: 0x00 ..
OK
Getting VTARGET.. 4.0V .. OK

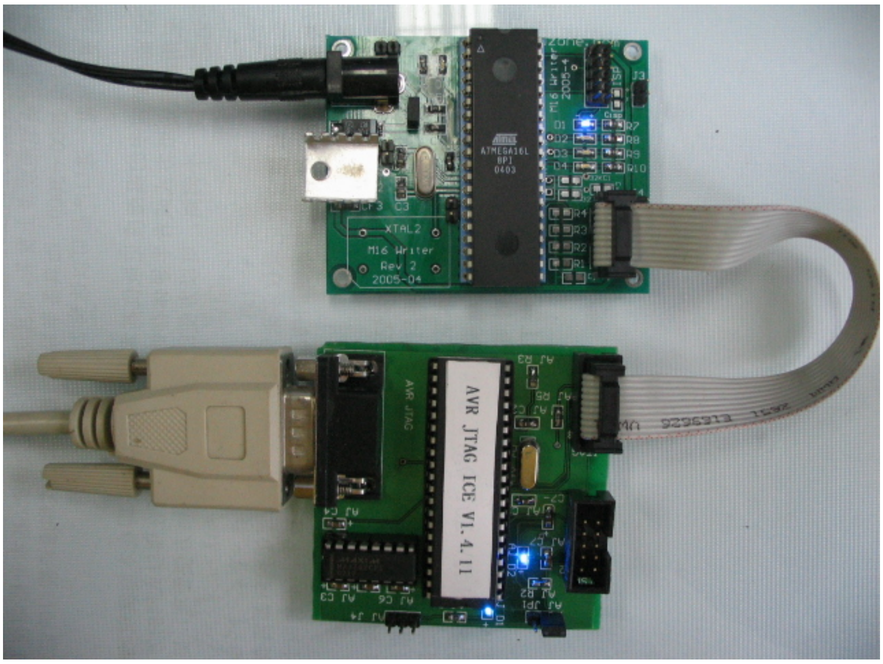
第三章 实战练习

本章通过一个实际例子来演示各个版本的JTAG ICE的操作和跳线设置以及一些注意事项。

首先安装AVR STUDIO，本实例使用AVR STUDIO 4.11+SP3；然后安装AVR C编译器WINAVR，本实例使用20050214版本；接着按照以下使用指南连接JTAG ICE和目标板，本实例采用了本站的M8/16 Writer作为目标板；最后准备源程序，本实例采用了一个LED_FLASH的例子，该例子可以在www.mcuzone.com下载。完成这些准备工作后即可以根据不同版本的JTAG进行3.1、3.2或者3.3的操作。

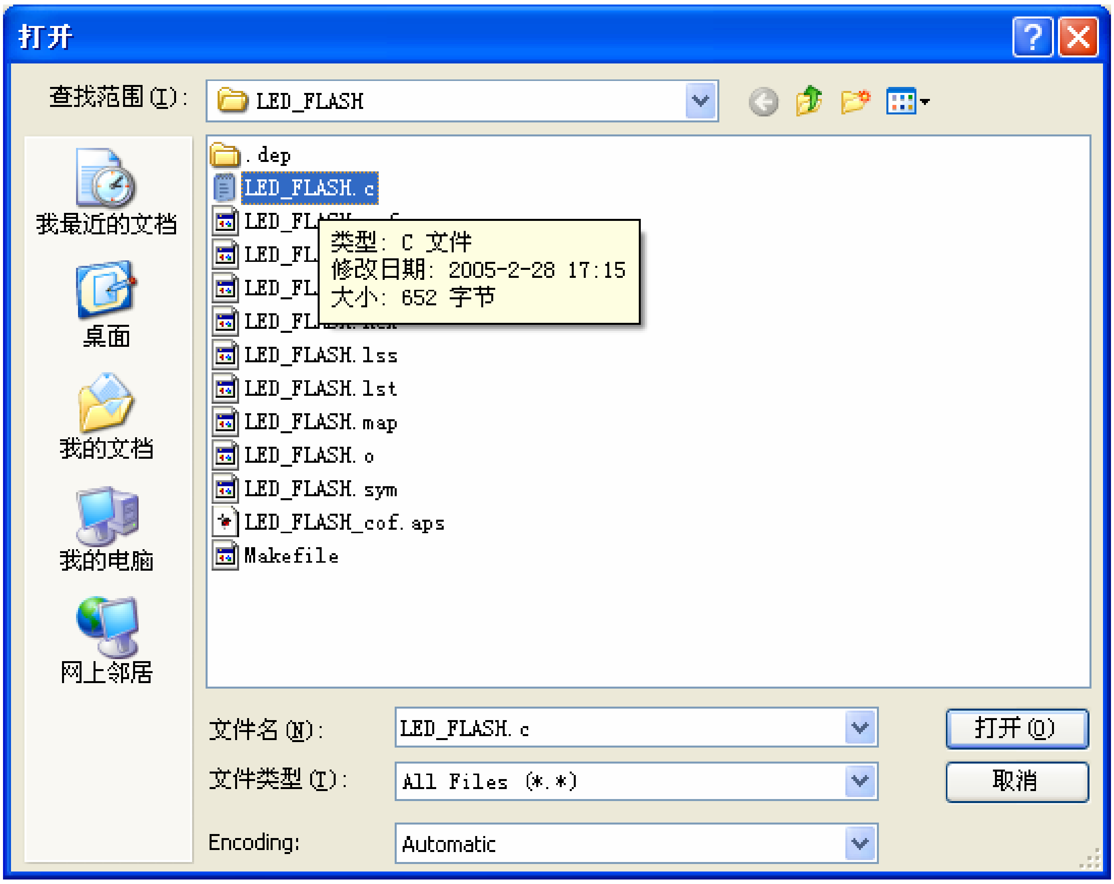
3.1 AVR JTAG ICE V1的使用指南：

1，按照下图所示进行硬件连接：

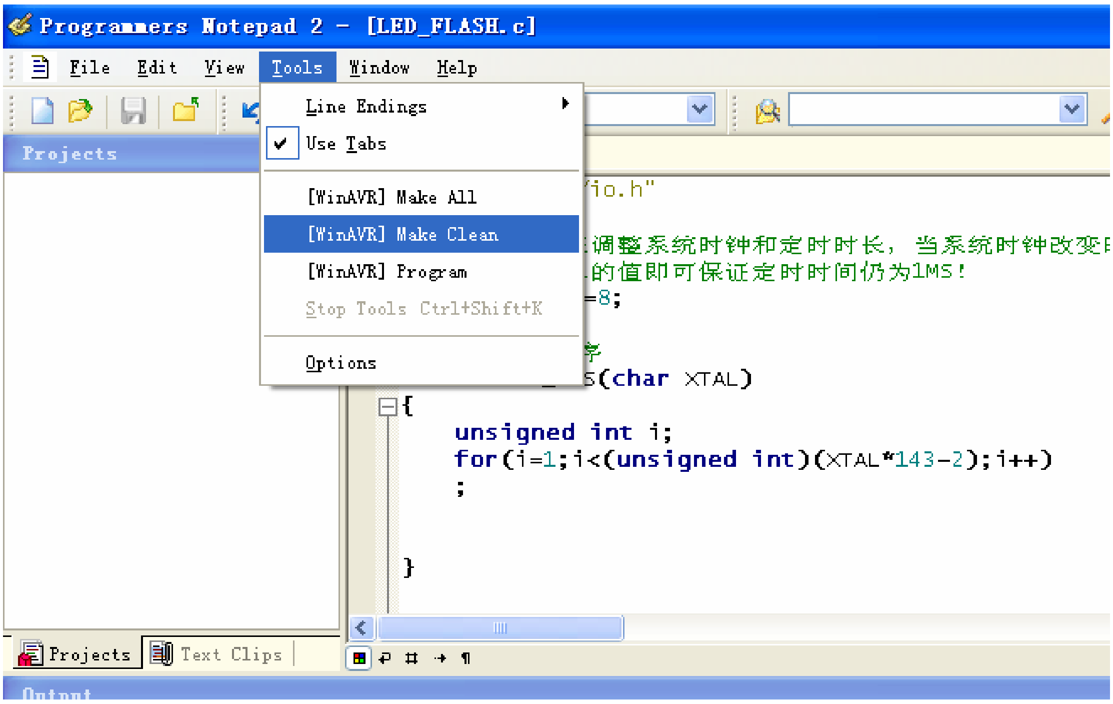


2，打开AVR STUDIO 4.11

本示例采用WINAVR20050214版本进行编译，打开WINAVR的PN(Programmers Notepad)，调入源程序，本示例使用了LED_FLASH例子，该示例已经有Makefile存在了（如果不使用本示例，请自行解决源程序和目标板的问题），打开LED_FLASH.c，请注意，后缀.c需要小写！！如下图：

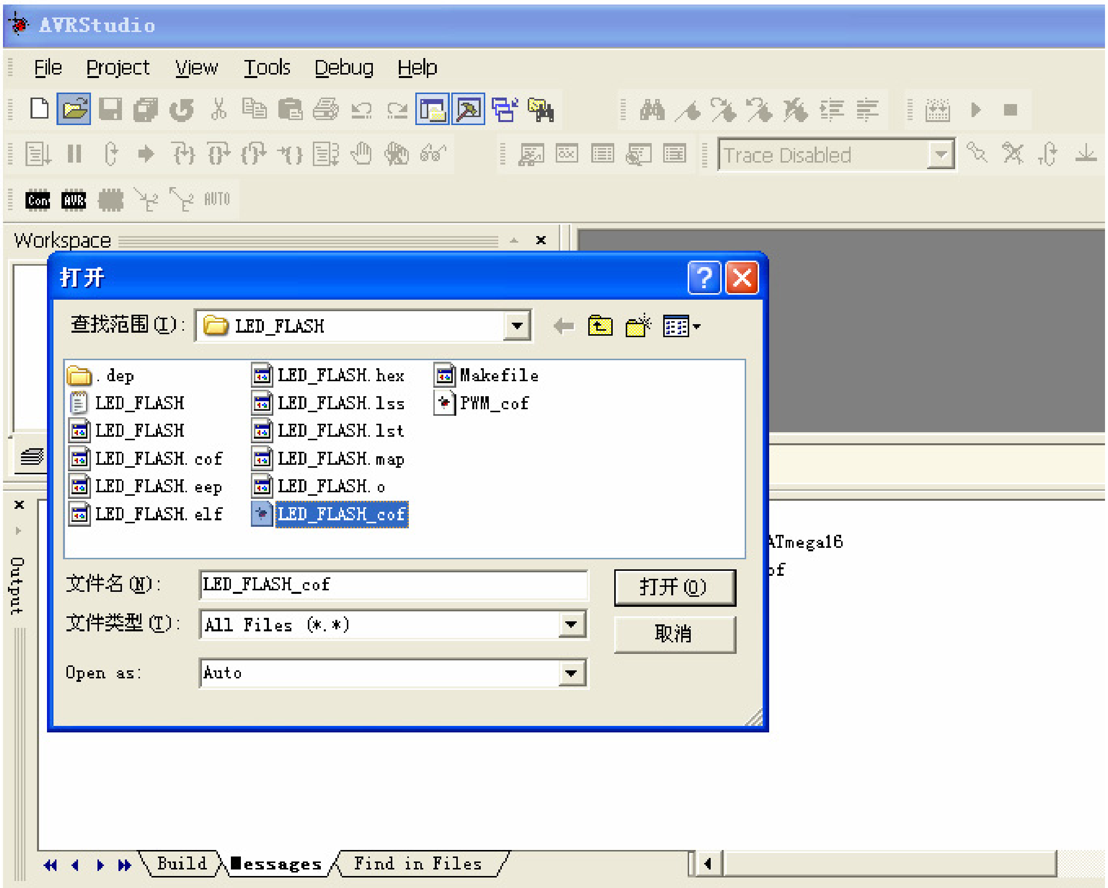


如果下载的示例的已经编译过的，那需要先清空，然后重新编译，相关选项可以参照下图，即先执行Tools下面的Make Clean，然后再执行Make All。

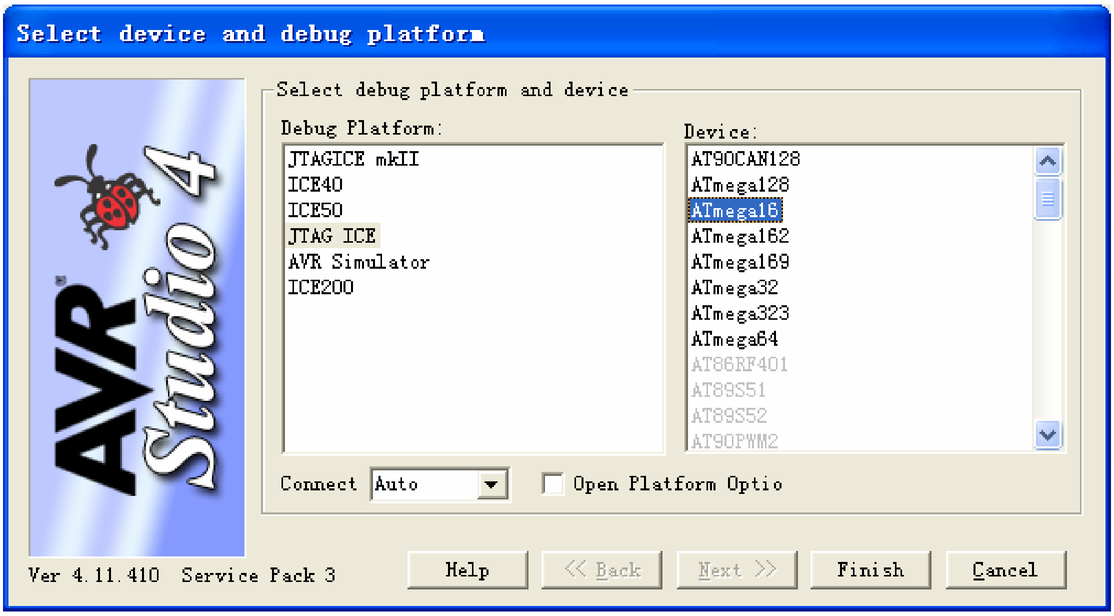


由于在Makefile里面已经设置好了，故只需Make All一下即可。编译通过后WINAVR会生成AVRStudio所需要的.cof文件。

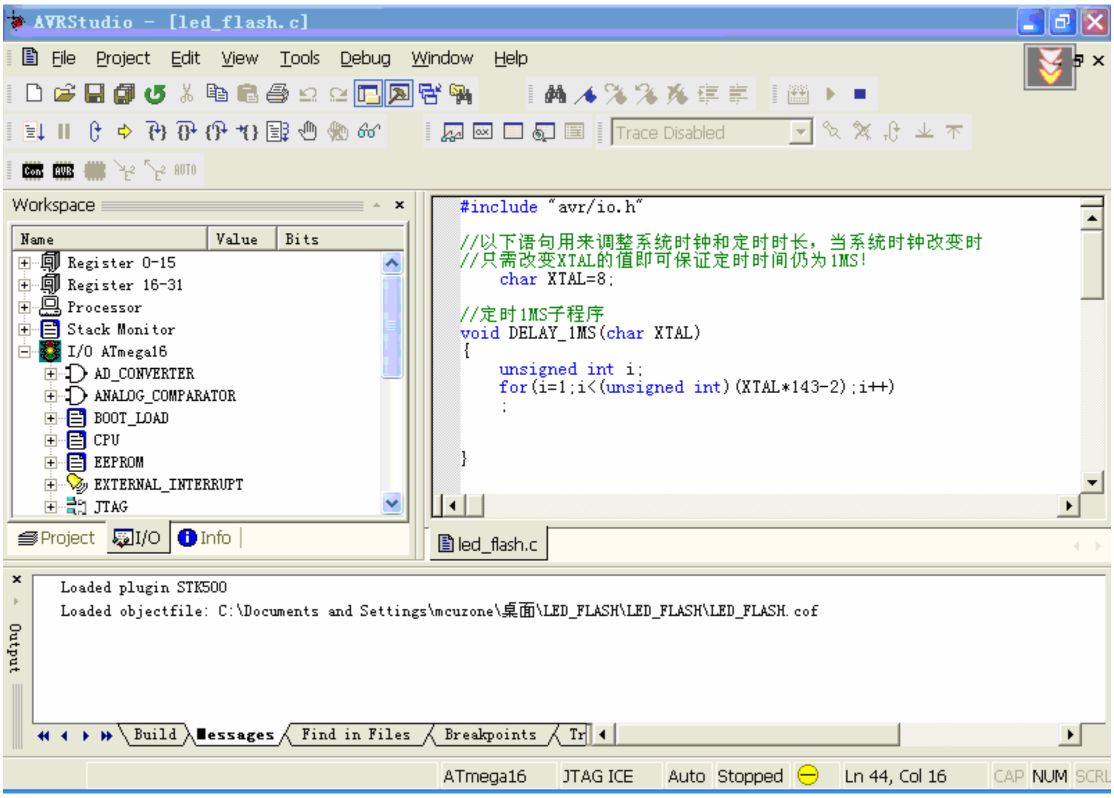
接下来打开AVRStudio，本示例采用了4.11版本；调入LED_Flash.cof，如下图所示：



调入.cof文件后AVRStudio会询问采用什么调试平台，使用什么目标芯片，通过那种接口，如下图所示：

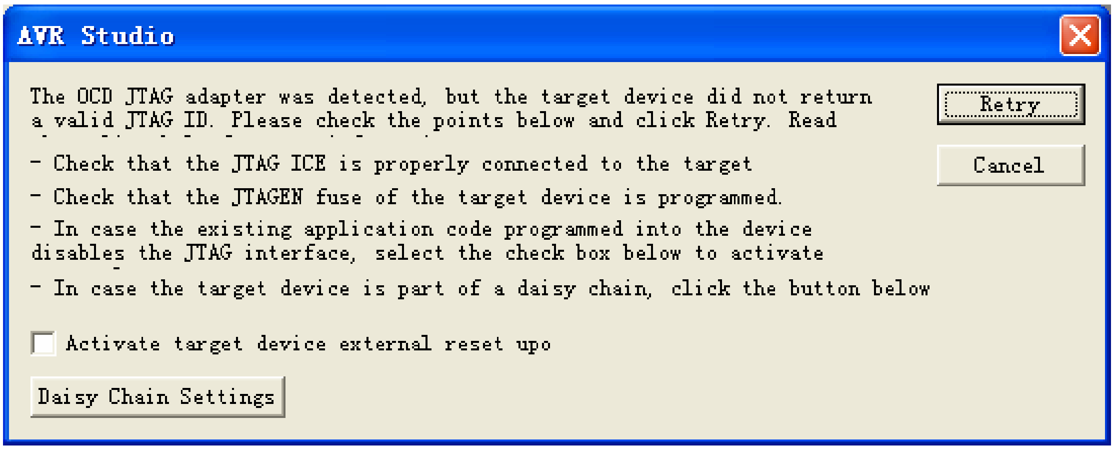


本示例应选择JTAG ICE和ATmega16，接口设置成Auto。点击Finish后出现下图所示内容的时候即表示JTAG ICE和目标板均工作正常。

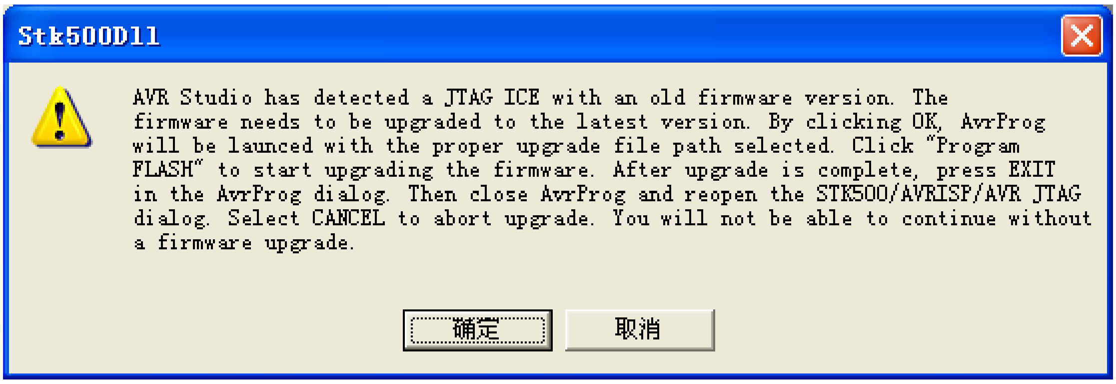


这时可以进行单步，全速，断点，查看寄存器等等操作.

如果点击Finish后并未出现上图所示内容，而是出现下图所示内容，则表示目标板未正确连接，请检查硬件连接并关闭一切占用串口的程序，然后再重新连接。



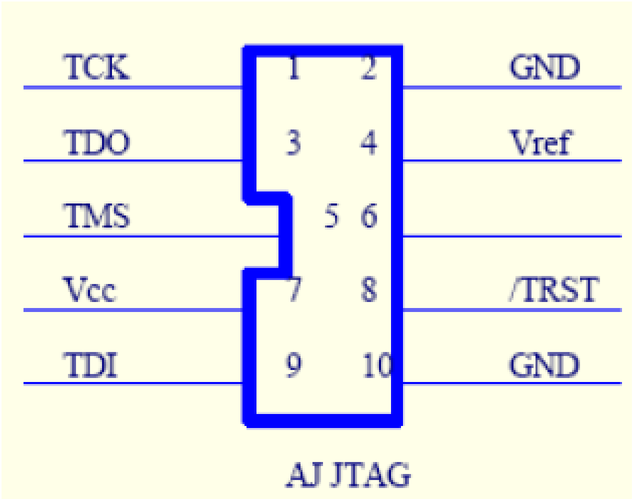
如果出现下图所示内容，则表示JTAG ICE的固件需要升级，用户可以按照Mcuzone的提示执行升级或者直接发回Mcuzone由专业人士为您免费升级！



3，注意事项：

本JTAG ICE 通过串口和PC 机相连接，在JTAG ICE上的串口为DB9 母口，通过一条非交叉的串口延长线与PC相连。

下图为V1版本的JTAG ICE的10芯引脚定义：



其中1 脚TCK、3 脚TDO、5 脚TMS、9 脚TDI 分别和具备JTAG 功能的AVR（以ATMEGA16 为例，分别为24 脚TCK、26 脚TDO、25 脚TMS、27 脚TDI）的相应引脚连接。Vcc 为JTAG 电源，用于给JTAG ICE 上的芯片供电；Vref 为目标板电源，2 和10 脚为电源地。/TRST 可不接。

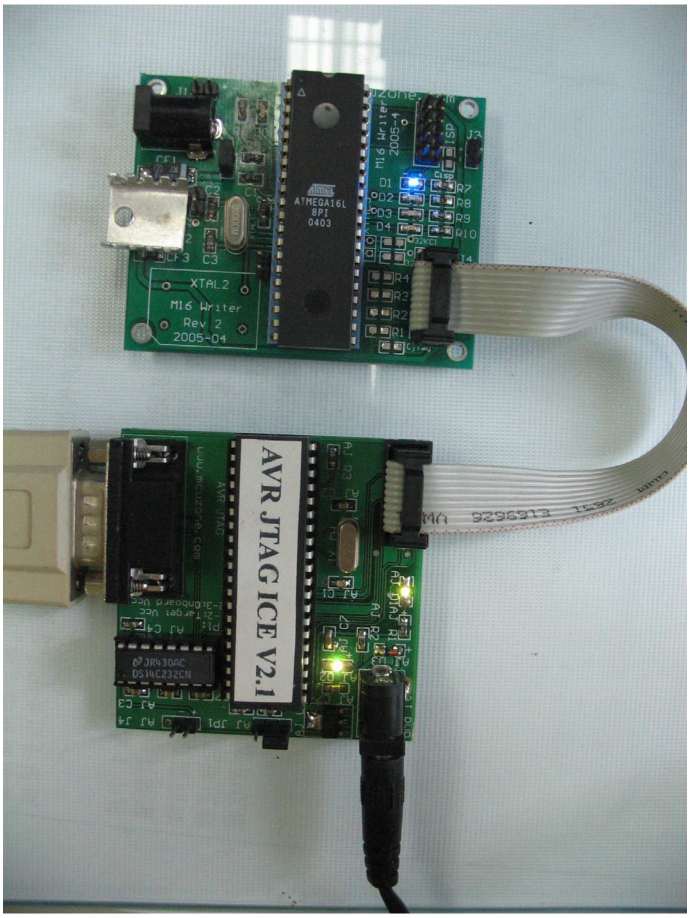
另外一个需要注意的接口是AVR JTAG ICE 上的AJ JP1，该跳线用来选择目标板和JTAG ICE 的电源，当跳线短接1－2 时，Vcc 和Vref 被短接，即JTAG ICE和目标板公用电源；当跳线短接2－3 时，Vcc 和Vref 独立供电，JTAG ICE 的电源由AJ J4供电，其中1 脚为Vcc，2 脚为GND，3 脚悬空（从左到右）。如果在无特殊电平要求情况下建议目标板和JTAG ICE 公用一组电源，即短接AJJP1的1-2，默认情况的跳线即为此连接。

注意：

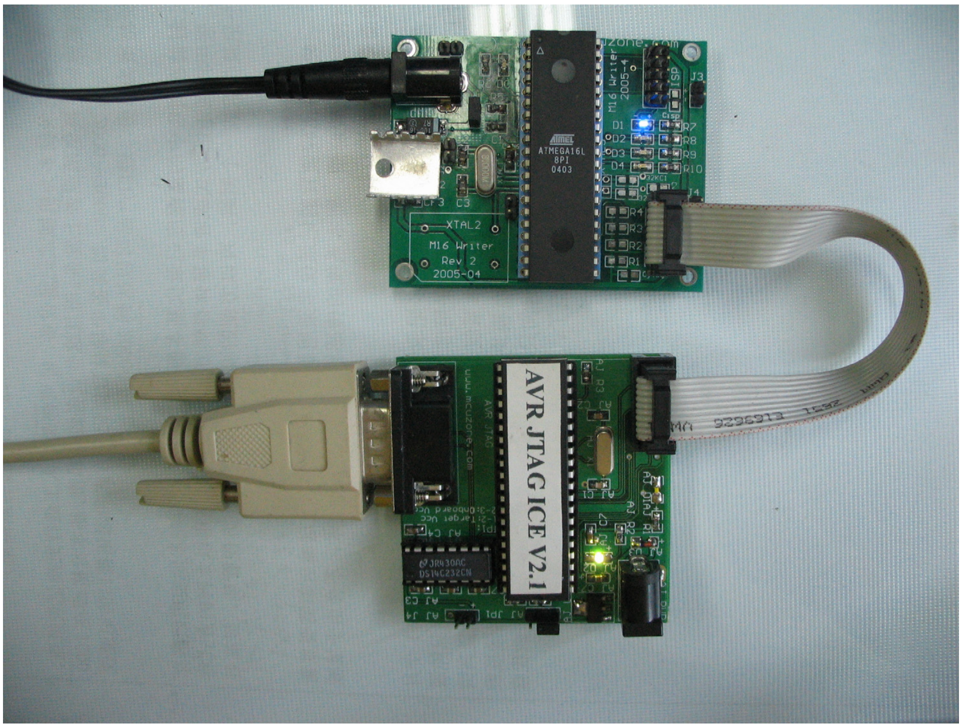
V1版本的JTAG ICE板载ISP接口，用户可以通过该接口自行升级JTAG ICE的固件。

3.2 AVR JTAG ICE V2的使用指南：

1，按照下图所示进行硬件连接：



JTAG供电，短接1－2



目标板供电，短接1-2

2，同V1。

3，注意事项：

主要是要注意JTAG和目标板的电源管理。强烈建议采用单一电源供电，如果将电源加在JTAG上，则跳线应设置成1-2短接以实现对目标板的供电；如果电源加在目标板上，则将跳线设置成2-3短接实现JTAG的取电。

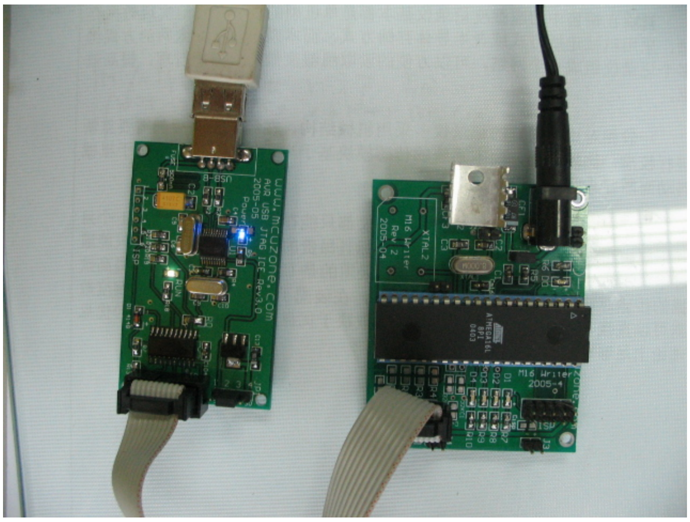
串口连接线为非交叉延长线！

注意：

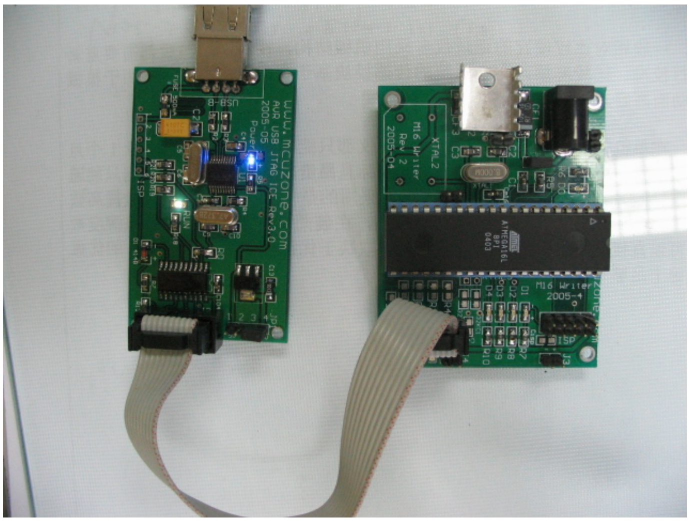
V2版本的JTAG ICE板载LDO，未经稳压的外部电源可以通过该LDO稳压后给系统供电，输入电压范围为5.5V-9V；如果外部电源为稳压过的3-5VDC，则也可以直接将该电压加到J4插针上。

3.3 AVR JTAG ICE V3的使用指南：

1，按照下图所示进行硬件连接：

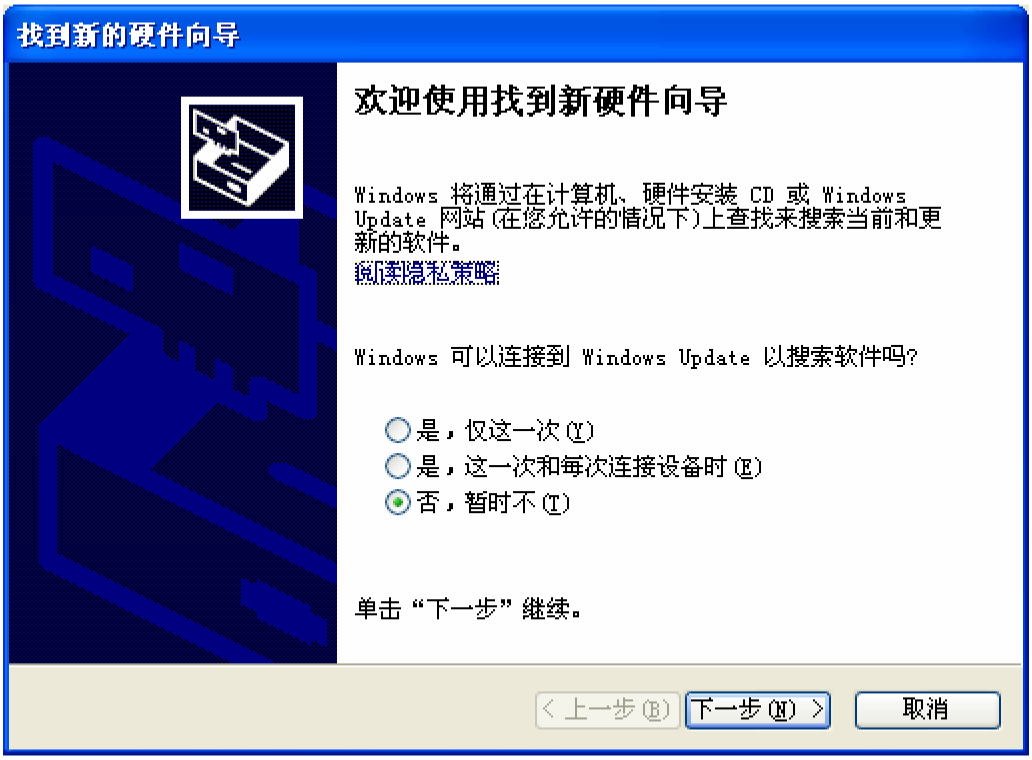


目标板独立供电，短接2－3

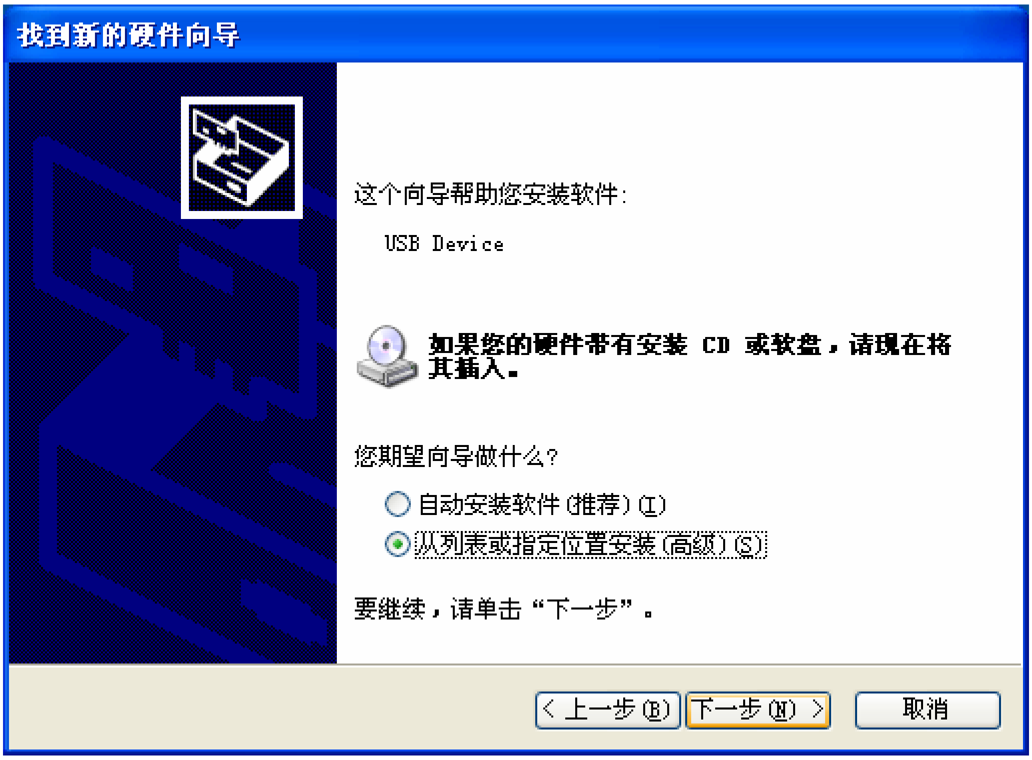


USB取电，短接1－2或者3－4

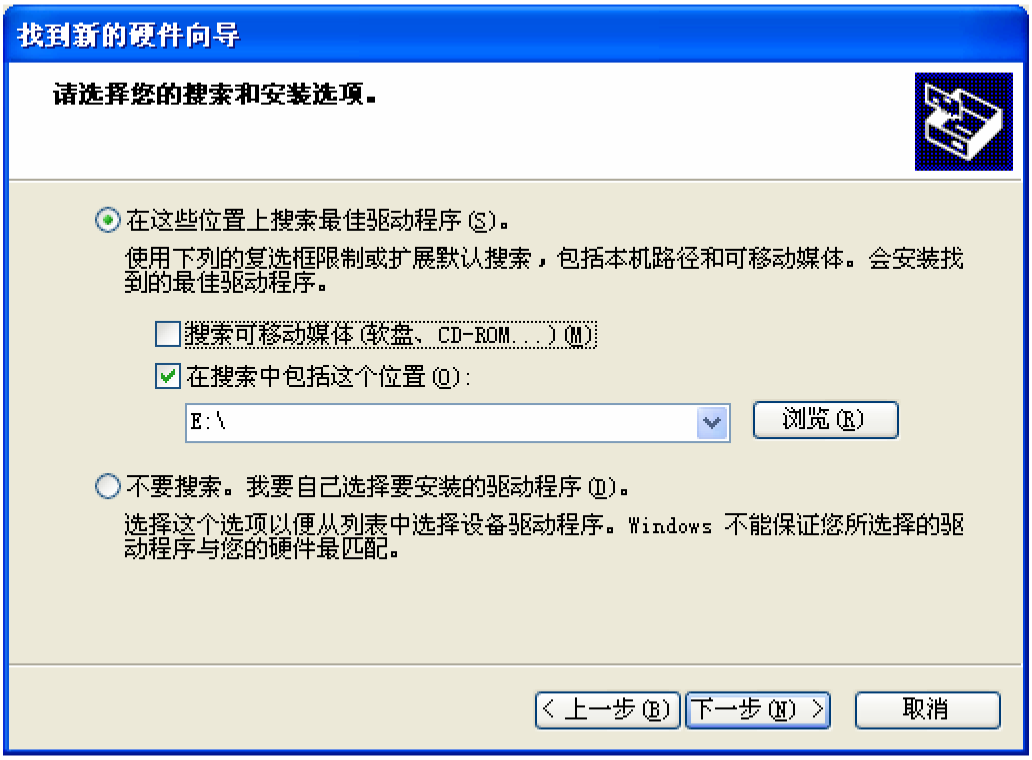
系统将提示找到新硬件，这时可以按照下图进行驱动程序的安装。



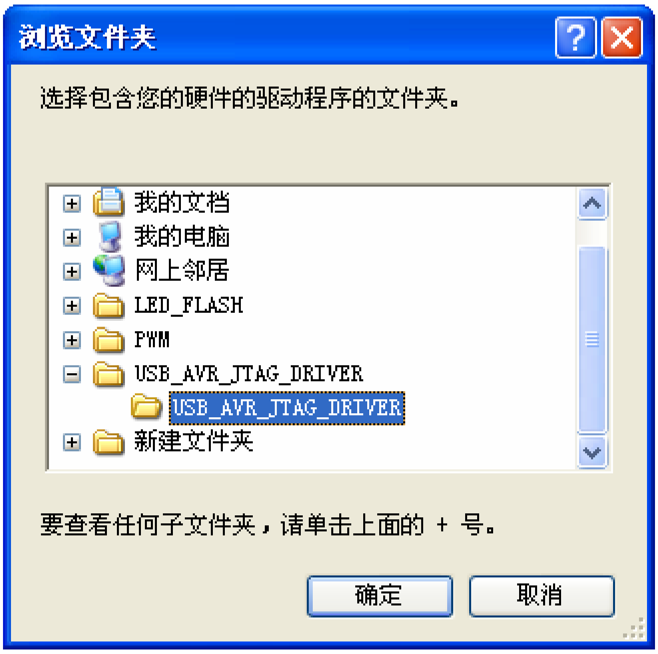
选择“否，暂时不”，然后点击“下一步”。



选择“从列表或指定位置安装”，然后点击下一步。
(也可以讲附赠光盘放入光驱，然后选择“自动安装软件”)



通过“浏览”按钮定位驱动程序所放置的位置，驱动程序可以在www.mcuzone.com下载到，也可以通过联系Mcuzone另行提供。



选择驱动程序所在位置，点击确定，系统开始安装驱动。

找到新的硬件向导
向导正在搜索，请稍候...
USB Device
< 上一步(B)
下一步(N) >
取消

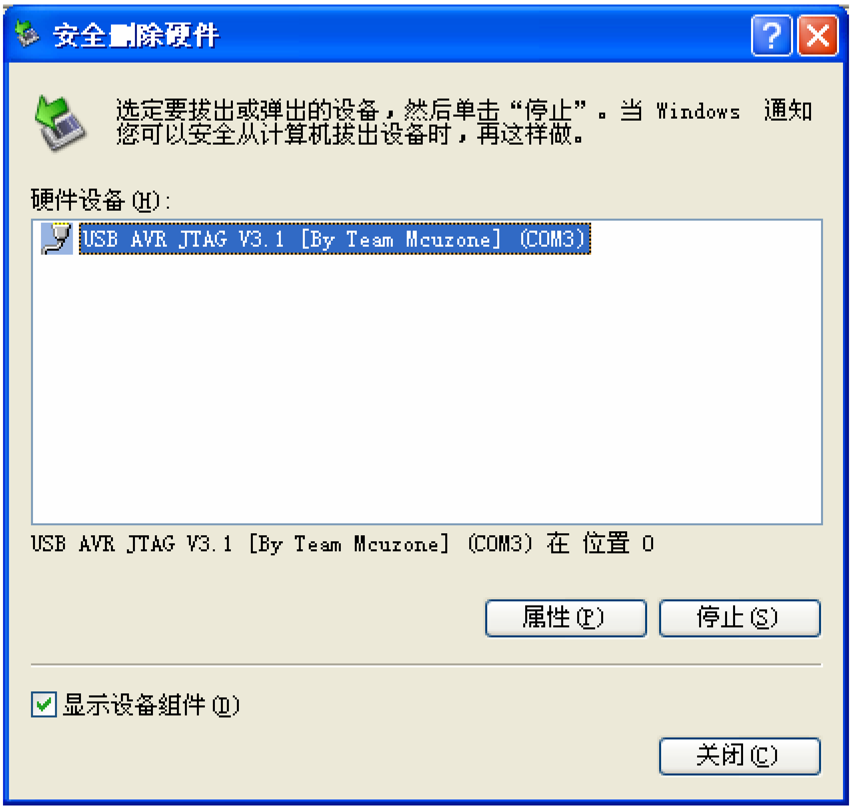
请稍等片刻。

找到新的硬件向导
向导正在安装软件，请稍候...
USB AVR JTAG V3.1 [By Team Mcuzone]
硬件安装
正在为此硬件安装的软件:
USB AVR JTAG V3.1 [By Team Mcuzone]
没有通过 Windows 徽标测试，无法验证它同 Windows XP 的相容性。(告诉我为什么这个测试很重要。)
继续安装此软件会立即或在以后使系统变得不稳定。Microsoft 建议您现在停止此安装，并同硬件供应商联系，以获得通过 Windows 徽标测试的软件。
仍然继续(C)
停止安装(S)

系统提示未通过徽标测试，不予理会，点击“仍然继续”。

安装进行中。

完成安装，点击完成，无需重启。



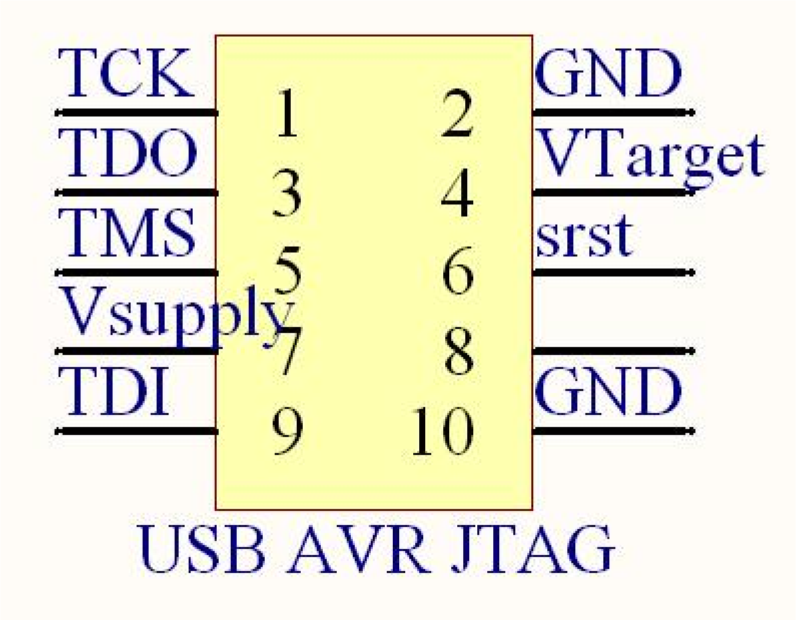
点击右下脚即插即用的图标，出现上图所示信息。

2，同V1操作。

3，注意：

对于USB版本的AVR JTAG ICE，有几点需要注意，首先，请注意JTAG ICE和目标板的电源管理，JTAG ICE的供电由USB端口提供，目标板的电源可以自行提供也可以通过USB取电。当目标板单独供电的时候请参照后文将USB AVR JTAG ICE上面的电源跳线跳在2-3；如果目标板从USB端口取电，且为3.3V则应将电源跳线跳在3-4，如果为5V，则应跳在1-2；其次如果更换USB端口的时候虚拟出来的串口的编号将会改变，如之前为COM3，更换USB端口后将变成COM4，所以更换USB口后请在AVRSTUDIO里面设置成Auto方式进行连接，或者手工设置新的虚拟COM端口；最后，由于本桥接芯片为即插即用类型（类似于U盘），所以如果使用完毕需要拔出，请先双击（或者左键单击）右下角即插即用图标进行安全删除硬件。

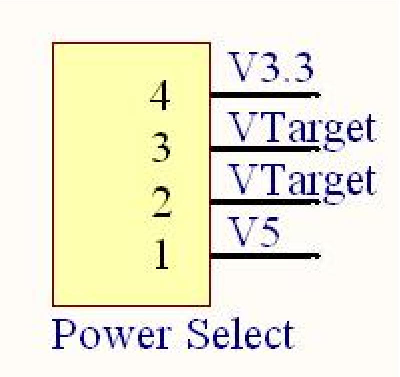
4，10芯JTAG引脚定义：



10 芯 JTAG 的引脚定义

其中 Vsupply 为目标板的供电电源，如果目标板和 USB JTAG 分开供电则需将电源跳线跳至 2-3；如果目标板从 USB JTAG 取电，且为 3.3V，则可以将电源跳线跳至 1-2；如果目标板需要 5V 电源则将电源跳线跳至 3－4。
注意：USB AVR JTAG 虽然进行了一定的隔离和电源保护措施，但是为了确保主机、USB JTAG 以及目标板的安全，请在上电之前检查一下电源电路！！！

5，4 芯电源跳线说明



4芯电源选择定义

跳线说明：
1-2：目标板采用USB端口供电，且供电电压为5V；
2-3：目标板独立供电；
3-4：目标板采用USB端口供电，且供电电压为3.3V；

附录一

特别注意事项

使用JTAG ICE 时，要考虑以下注意事项。

JTAG 端口频率

应该将JTAG 端口频率设置为1/4 或低于目标器件频率。当使用一个支持系统时钟分离的器件时，选择JTAG ICE 端口频率必须考虑最低系统频率。

IO 周边器件

即使程序执行到断点时被中断，所有的IO 设备也将继续运行。例如：当程序执行到UART 传输时被断点中断，UART 将完成传输，并将TXC(传输结束)标志位置位，以及取得下一个单步执行的代码。

在停止模式下所有的IO 模块都会继续执行，除了下面列出的两个特殊模块：

1. 定时器/计数器

   在调试选项页中，用户可以设置在停止模式时定时器/计数器是否继续执行。

2. 看门狗定时器

   在停止模式下，看门狗定时器将会终止运行。这避免了终止模式下的无效看门狗复位。

单步执行

因为IO 在停止模式下继续运行，所以应该考虑到在IO 执行时单步操作对IO 的延时。

例：

OUT PORTB , 0xAA

IN TEMP, PINB

在器件中直接运行程序，IN 指令无法读回0xAA 这个值。为确保在引脚寄存器出现正确数值，应该在OUT 和IN 指令间加一条NOP 指令。然而，在JTAG ICE 中单步执行代码时，单步执行会在PIN 寄存器中读到0xAA 。这是因为IO 口没有停止，并以全速继续运行，这给数据锁存到引脚并回送提供了充足的时间。

软件断点

在JTAG ICE 中，某些AVR 器件支持中断指令。使用中断指令时，应该注意以下事项：因为软件断点取代了原始的中断指令，每次增加或删除软件断点时，都要对Flash 重新编程。AVR FLASH 只有1000 次的擦写周期。大量使用中断指令将会最终损坏FLASH。而且重新编译后的HEX 中包含更多的断点信息，因此也会在仿真时引入更多的启动延时。

注：在调试阶段，需要频繁的改变断点位置，应避免使用软件中断指令，建议使用硬件断点。这会提高速度，延长器件的使用期限。

调试中目标AVR 的复位和掉电

如果调试时JTAG ICE 突然断电，通信将会失败。如果只有目标AVR 断电，还可能继续调试。然而，当目标AVR 复位或掉电时，目标AVR 可能会进入另一种状态。然而，硬件复位后，硬件断点还依然保存在目标AVR 中。

注：若要使调试恢复正常，请先执行“Debug->Stop debugging”菜单操作，再执行“Debug->Start debugging”进入调试模式。

与JTAG 相关的熔断丝设置

为正确运行，目标器件必须设置两个熔丝，即JTAG 使能熔丝和OCD 使能

熔丝。当芯片出厂时，这两个熔丝已被默认激活。如果你无意中关闭了JTAG 使能熔丝，你能通过通用编程器或ISP 来重新激活该熔丝。每次调试时，AVR Studio 都能自动激活OCD 熔丝，退出调试时AVR Studio 会恢复其原有设置。

信息窗口

信息窗口提供了AVR Studio 和JTAG ICE 运行时的重要信息。

使用JTAG ICE 调试Atmega128

这部分介绍仅限于Atmega128。当用Atmega128 执行中断指令时，应该注意以下事项：假设在代码中有条中断指令，指令在断点处刚好执行。当从这个位置单步执行时，将会延时一个额外的周期才发生中断。可能这种行为并不是至关重要，但如果有多条连续的中断指令，那么程序将在连续的中断指令执行完毕后才会发生中断。所以，当有意要使用这样的功能时，建议使用硬件断点。

访问IO寄存器

如果没有改变微处理器的状态，某些寄存器将不能被读取。但是改变微处理器的状态将使得一些状态位被置位或清除，更甚者，将会破坏某些寄存器的内容，从而导致更多的仿真错误。因此，下面的寄存器不能直接使用JTAG ICE 进行访问：

ATmega323	UDR, SPSR, UCSRC/UBRRH, TWCR
ATmega16	
Atmega64 Atmega128	UCSR0A, UDR0, SPSR, SPDR, ETIFR, TWCR
ATmega162	UCSR1A, UDR1, ACSR, UCSR0A, UDR0, SPDR, TIFR, GIFR, ETIFR: No write access
	UDR1, UDR0, SPDR: No read access
ATmega169	PINA, PINB, PINC, PIND, PINE, PINF, PING, TIFR0, TIFR1, TIFR2, EIFR, SPDR, ACSR, USISR, UCSR0A, UDR0, LCDCRA: No write access
	SPDR, UDR0: No read access
ATmega32	ACSR, UCSRA, UDR, SPDR, TWCR, TIFR, GIFR: No write access
	UDR, SPDR: No read access

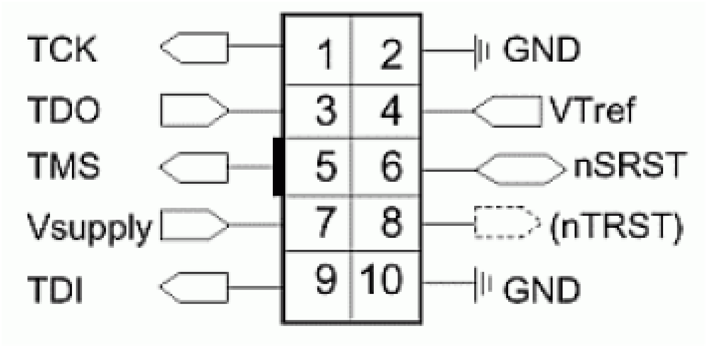
JTAG引脚功能

如果JTAG 接口被激活，JTAG 引脚就不能当作通用IO 引脚使用。但是如果在程序代码中将MCUCSR 中的JTD（JTAG Disable）位打开，或者用编程器将JTAG 使能熔丝关闭，JTAG接口就可以当做一般IO 口使用。否则JTAG 接口只能用于JTAG 功能。

校验Flash

当使用JTAG ICE 调试时，有可能需要对Flash 编程。注意：为了优化速度，JTAG ICE没有在重编程后对FLASH 做校验，除非用户指定校验选项。AVR 闪存允许10000 次的擦写操作。所以多次的编程操作最终将损坏闪存。

附录二 JTAG 连接器的引脚定义及说明



JTAG 信号

PIN	信号	I/O	描述
1	TCK	Output	测试时钟，从 JTAG ICE 到目标 JTAG 端口的时钟信号
2	GND	-	地
3	TDO	Input	测试数据输出，从目标 JTAG 端口到 JTAG ICE 的数据信号
4	Vtref	Input	目标参考电压。VDD：目标板电源电压
5	TMS	Output	测试模式选择，由 JTAG ICE 到目标 JTAG 端口的模式选择信号。
6	nSRST	OUT/Input	由适配器到目标系统复位的开放集电极输出。这个引脚也是适配器的输入，将目标初始化信息上送给 JTAG ICE
7	Vsupply	Input	JTAGICE 的电源电压。
8	nTRST	NC(Output)	不被连接，兼容其他设备（JTAG 端口复位）
9	TDI	Output	从 JTAG ICE 到目标 JTAG 端口的测试数据输入、数据信号。
10	GND	-	地

附录三 版本信息

Rev3.2:

2005-7-10更新，修订以下内容：

1，添加了设置多余3个断点后的错误信息图片

2，JTD可以用ISP打开

3，WinAVR的输出也支持变量名断点

Rev3.1:

2005-7-5更新，修订以下内容：

1，修改图片，去除无用信息

2，加入错误提示截图

3，加入硬件连接图片

Rev3.0:

2005-7-3更新，将V1和V2版本的AVR JTAG ICE说明书更新，加入V3版本的AVR JTAG ICE使用说明，形成Rev3.0使用说明的原始版本。

感谢你的关注！

如果有任何问题请直接联系：

hotislandn@hotmail.com

hdapple_2000@hotmail.com

如果发现文档的错误或有更好的建议，也请联系，以便我们加以改进，在此先表示感谢！

请经常访问www.mcuzone.com以获取最近更新。